ALGORITMO DE CLASSIFICAÇÃO POR PARTICIONAMENTO HIERÁRQUICO

Lygia Marina Mendes da Costa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Carlos Eduardo Pedreira

Rio de Janeiro
Novembro de 2017

ALGORITMO DE CLASSIFICAÇÃO POR PARTICIONAMENTO HIERÁRQUICO

Lygia Marina Mendes da Costa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Carlos Eduardo Pedreira, Ph.D.

Prof. Rodrigo Tosta Peres, D.Sc.

Prof. Valmir Carneiro Barbosa, Ph.D.

Prof. Guilherme de Alencar Barreto, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2017

da Costa, Lygia Marina Mendes

Algoritmo de Classificação por Particionamento Hierárquico/Lygia Marina Mendes da Costa. – Rio de Janeiro: UFRJ/COPPE, 2017.

XIV, 57 p.: il.; 29, 7cm.

Orientador: Carlos Eduardo Pedreira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2017.

Referências Bibliográficas: p. 39 – 41.

1. Classificação local. 2. Particionamento hierárquico. 3. Divergência de Cauchy-Schwarz. I. Pedreira, Carlos Eduardo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À família.

Agradecimentos

Agradeço aos Professores Carlos Eduardo Pedreira e Rodrigo Peres pela paciência e disponibilidsde. Obrigada, também, à banca examinadora por aceitar o convite e se dispor a participar da defesa desta dissertação.

Por fim, sou grata a todos que de alguma forma contribuíram com a minha formação e me auxiliaram em toda a minha jornada acadêmica.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ALGORITMO DE CLASSIFICAÇÃO POR PARTICIONAMENTO HIERÁRQUICO

Lygia Marina Mendes da Costa

Novembro/2017

Orientador: Carlos Eduardo Pedreira

Programa: Engenharia de Sistemas e Computação

Esta dissertação propõe um novo método de classificação por particionamento hierárquico que visa não só retornar uma resposta referente à classe de um elemento, mas também fornecer maiores informações quanto ao processo de classificação e quanto a disposição espacial das classes ao longo do espaço de atributos. Através de um particionamento iterativo e do uso de conceitos como divergência entre distribuições, o método busca encontrar regiões em que haja uma classe predominante e regiões em que a sobreposição entre as classes torna a classificação mais complexa. Experimentos com bancos de dados artificiais e reais foram realizados para demonstrar a competitividade do método e a sua vantagem em separar regiões de fácil classificação de regiões mais complexas, tanto para classificação própria quanto para obter maiores informações quanto ao desempenho de outros métodos mais conhecidos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

HIERARCHICAL PARTITIONING ALGORITHM FOR CLASSIFICATION

Lygia Marina Mendes da Costa

November/2017

Advisor: Carlos Eduardo Pedreira

Department: Systems Engineering and Computer Science

This dissertation proposes a new method of hierarchical partitioning classification that aims not only to return a response regarding the class of an element, but also to provide more information about the classification process and the spatial arrangement of the classes along the attribute space. Through iterative partitioning and the use of concepts such as divergence between distributions, the method seeks to find regions where there is a predominant class and regions where overlap between classes makes classification more complex. Experiments with artificial and real databases were performed to demonstrate the competitiveness of the method and its advantage in separating regions of easy classification of more complex regions, both for own classification and to obtain more information on the performance of well-known methods.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACPH	algoritmo de classificação por particionamento hierárquico, p. 3, 23–27, 31, 32
RBFNN	rede neural de funções de base radial, p. 1, 2
SVM	máquina de vetores de suporte, p. 2, 4, 7, 23–26, 31, 32
VQ	quantização de vetores, p. 6
kNN	k-vizinhos mais próximos, p. 1–4, 6, 23–26, 31, 32

Capítulo 1

Introdução

Classificação é um processo amplamente aplicado em diversas áreas de estudo. Problemas como diagnóstico de doenças [1], reconhecimento de objetos em imagens [2] e categorização de textos e periódicos [3, 4] são alguns dos exemplos mais comuns. Classificar, portanto, consiste em mapear observações de um dado fenômeno a um conjunto discreto e finito que representa todas as classes possíveis da aplicação [5].

Os métodos de classificação estão inseridos em uma lógica local-global. Uma abordagem puramente global pressupõe que as observações são geradas por um fenômeno regido por uma lei fundamental geral e não considera eventuais relações locais [6]. Assim, modelos globais visam representar o conjunto de dados como um todo. Em contrapartida, modelos locais partem do pressuposto de que diferentes subconjuntos da amostra são regidos por diferentes distribuições. Logo, enquanto métodos globais levam em conta toda a amostra, abordagens locais usufruem de características pontuais para classificar subconjuntos independentemente.

Um método local bem conhecido é o k-vizinhos mais próximos (k-nearest neighbors, kNN) [5, 7, 8], que consiste em considerar o rótulo dos k vizinhos mais próximos da observação em teste a fim de decidir a qual classe atribuí-la. Nesse caso, a classe estimada será a mais frequente entre os vizinhos. Embora o kNN seja um algoritmo simples e, em geral, com bom desempenho na prática, ainda é muito sensível à escolha da vizinhança e a ruídos [9]. Nesse sentido, diversas pesquisas propõem novos métodos baseados no kNN a fim de contornar suas limitações. GOU *et al.* [9], por exemplo, introduziu uma nova medida de proximidade para diminuir a influência de vizinhos discrepantes, enquanto PAN *et al.* [10] propôs que os k vizinhos mais próximos fossem selecionados de forma que todos possuam uma vizinhança compartilhada com a observação de teste. Ambas as modificações visam tornar o kNN menos sensível à escolha da vizinhança. Já no artigo apresentado por CALVO-ZARAGOZA *et al.* [11], o algoritmo foi incrementado com uma fase de pré-seleção de protótipos a fim de reduzir o custo computacional e a sensibilidade a ruídos.

Outra ferramenta local de classificação semelhante ao kNN é rede neural a base

de funções radiais (radial basis function neural network, RBFNN) [5, 12, 13]. Nele, os neurônios intermediários são responsáveis por verificar a similaridade entre observação e protótipos. Assim, o resultado parcial de cada neurônio representa a pertinência da observação à classe do protótipo em questão e a classificação final é dada pela classe mais frequente entre esses resultados parciais.

Por outro lado, alguns métodos se apresentam tanto abordagens locais quanto globais. Máquina de vetores de suporte (support vector machine, SVM) [5], por exemplo, possui um caráter global na medida em que seu principal objetivo é gerar um único hiperplano como fronteira entre duas classes distintas. No entanto, considera apenas um subconjunto de observações para ajustar os parâmetros desse hiperplano, fazendo com que o método também seja visto como um algoritmo local. Entretanto, diferentemente do kNN e do RBFNN, o SVM não considera diretamente a vizinhança local da observação teste para classificá-la, mas sim a relação entre a fronteira e o ponto avaliado.

Todos os algoritmos mencionados até aqui estão inseridos no conjunto de métodos de aprendizado supervisionado [7]. São algoritmos que consideram observações previamente classificadas para decidir como classificar uma nova entrada. Entretanto, há casos em que a aplicação não oferece essa informação ou considerá-la não é apropriado, sendo necessário o uso de métodos não-supervisionados [7]. Em geral, os algoritmos não-supervisionados utilizados para classificação procuram agrupar observações semelhantes ou separar aquelas que são muito discrepantes. Diversas são as metodologias de agrupamento e de cálculo de semelhança, mas uma ferramenta amplamente utilizada nesses casos é o k-médias (k-means) [5, 8]. Trata-se de um algoritmo iterativo que visa dividir um conjunto de observações em k subconjuntos de modo que cada observação seja inserida no subconjunto com a média mais próxima.

Diversos são os trabalhos que visam aprimorar o uso do k-médias, principalmente quando aplicado a amostras de alta dimensão. CHENPING HOU *et al.* [14] propôs um framework em que k-médias e algoritmos de redução de dimensionalidade são aplicados alternadamente no processo de agrupamento de observações. Em ZHANG *et al.* [15], embora seja proposto algo similar, o algoritmo de agrupamento é alternado com algoritmos de redução de dimensionalidade não-linear, o que permite, muitas vezes, o uso de reduções mais fieis aos dados originais.

Procurando aproveitar as vantagens de ambas abordagens, alguns trabalhos apresentam métodos locais-globais que combinam tanto etapas supervisionadas quanto etapas não-supervisionadas. PERES e PEDREIRA [6] utilizam aprendizado não-supervisionado para particionar os dados de entrada em regiões locais para, então, aplicar esquemas supervisionados baseados na regra de Bayes [8] em cada região independentemente. Em LEITE [16], o k-médias foi utilizado em um processo iterativo e hierárquico a fim de particionar a amostra para classificação local sem a

necessidade de se estabelecer um número de grupos k. Nesse caso, a classificação é feita segundo a classe mais frequente dentro da região, o que aproxima o método de LEITE [16] do kNN, uma vez que o método cria iterativamente, em sua essência, vizinhanças.

O uso do k-médias, ou qualquer outro algoritmo de agrupamento, como pré-processamento para a aplicação de métodos supervisionados locais permite alcançar grupos de observações que sejam de fácil classificação ou até grupos homogêneos, em que todos os dados pertencem à mesma classe. Paralelamente, o mesmo processo também possibilita alcançar regiões de difícil classificação, permitindo que se perceba os subconjuntos da amostra para os quais o método pode ser considerado insuficiente. Um dos principais benefícios dessas propostas é a possibilidade de determinar sub-regiões em que o classificador tem maior (ou menor) probabilidade de estimar a classe correta. O conhecimento a respeito dessas sub-regiões é de grande importância para aplicações mais sensíveis aos erros de classificação, em que transparência quanto ao processo de classificação é altamente recomendado. Um diagnóstico médico errado, por exemplo, é inadmissível, logo a medicina exige métodos que permitam que esse erro preditivo seja minimizado pelo auxílio de um especialista. Assim, o método de classificação passa a não mais estimar apenas rótulos referentes às observações, mas também a permitir melhor interpretação de seus resultados.

Nesse sentido, essa dissertação propõe um novo algoritmo de classificação baseado em particionamento hierárquico (ACPH) que explora a interpretabilidade e que procura fornecer um processo de aprendizado e de classificação mais confiante. O método aqui proposto se utiliza tanto do aprendizado supervisionado quanto do aprendizado não-supervisionado e estima não só a classe de determinada observação, mas também a confiança local desse resultado para cada dado de entrada. O particionamento hierárquico dispensa a necessidade de se especificar a priori o número de regiões em que a amostra será dividida e compõe a etapa não-supervisionada do algoritmo. Além disso, informações locais e globais são utilizadas em conjunto para decidir a continuidade do seu processo iterativo.

1.1 Contribuições

As contribuições desta dissertação estão resumidas conforme abaixo:

- Elaboração de um novo método de classificação local utilizando particionamento hierárquico e estimação de divergência entre distribuições.
- Desenvolvimento de um novo modelo de comparação entre algoritmos de classificação considerando o particionamento hierárquico.

- Elaboração de experimentos que demonstram o ganho de interpretabilidade fornecido pelo algoritmo proposto e que comparam o algoritmo com métodos mais comuns.
- Discussão a respeito do uso conjunto do algoritmo proposto e outros algoritmos de classificação.

1.2 Organização da Dissertação

Esta dissertação está organizada em cinco capítulos, incluindo este capítulo introdutório. No Capítulo 2 2, é feita uma revisão bibliográfica tratando de alguns algoritmos de classificação conhecidos, bem como outros dois algoritmos mais recentes. No Capítulo 3, é descrito o algoritmo proposto e os conceitos necessários para sua elaboração, bem como a configuração dos experimentos realizados. No Capítulo 4, são apresentados os resultados e a discussão proveniente da comparação entre o método proposto e métodos mais conhecidos. Por fim, o Capítulo 5 conclui o trabalho e discute trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Para o desenvolvimento e a execução desta dissertação, foi necessária a compreensão de alguns algoritmos relacionados a classificação e a particionamento de observações. Nesse sentido, as seções 2.1 e 2.2 descrevem dois algoritmos de classificação conhecidos que foram utilizados durante os experimentos desta dissertação. Já a seção 2.3 descreve dois algoritmos semelhantes ao proposto aqui e que serviram como base teórica para o trabalho.

2.1 K-Vizinhos Mais Próximos

Talvez um dos algoritmos mais simples de classificação, o k-vizinhos mais próximos (k-*nearest neighbors*, kNN) é um método supervisionado [5, 8] que dispensa fase de treinamento, uma vez que todos os cálculos necessários à classificação são dependentes da observação de teste.

Suponha um conjunto de observações $\mathbf{x} = \{x_1, ..., x_n\}$, em que um rótulo de classe $C \in \{A, B\}$ conhecido está associado a cada uma delas, e uma observação de teste x_j cuja classe não é conhecida. A regra de vizinhos mais próximos [8] diz que a classe de x_j pode ser estimada segundo a classe dos vizinhos x_i mais próximos dessa observação. Assim, seja $d(x_i, x_j)$ a distância entre x_i e x_j, se k vizinhos forem selecionados, tal que $n_k(A)$ e $n_k(B)$ sejam as quantidades de vizinhos com classe A e de vizinhos com classe B respectivamente, tem-se que a classe de x_j é dada pela classe majoritária entre os vizinhos, conforme abaixo.

$$r\acute{o}tulo(x_j) = C \quad | \quad n_k(C) = \max(n_k(A), n_k(B)) \quad e \quad C \in \{A.B\} \tag{2.1}$$

2.2 Máquinas de Vetores de Suporte

Máquina de Vetores de Suporte (support vector machine, SVM) é um método que pode ser utilizado com diversas finalidades, entre elas classificação e regresão [5, 8]. A ideia do SVM parte do pressuposto de que todo conjunto de dados constituído por duas classes, quando mapeado segundo uma transformação não-linear, pode ser separado por um hiperplano [8]. Assim, o método consiste em estimar a fronteira que melhor separa os dois subconjuntos de classes, isto é, procura o hiperplano que possui maior distância em relação às observações mais próximas desse hiperplano.

Suponha um conjunto de observações $\mathbf{x} = \{x_1, ..., x_n\}$, em que uma classe $C \in \{-1, 1\}$ está associada a cada uma delas. Seja uma transformação $\phi(.)$ tal que $\phi(\mathbf{x})$ resulta em um conjunto de dados $\mathbf{x}_\phi$ linearmente separáveis. Qualquer hiperplano que separa as duas classes pode ser dado por

$$y(\mathbf{x}) = w^T \phi(\mathbf{x}) = 0 \tag{2.2}$$

em que w^T determina a angulação do hiperplano [5]. é possível demonstrar por geometria que a distância perpendicular de uma observação $\phi(x_i)$ ao hiperplano é dada por $|y[\phi(x_i)]|/\|w\|$. Suponha que $y[\phi(x_i)]$ represente corretamente as classes de todas as observações, pode-se dizer que

$$C_i y[\phi(x_i)] \geq 1, \quad i = 1, ..., n \tag{2.3}$$

Então, multiplicando a expressão da distância entre ponto e hiperplano por C_i, dispensa-se o uso do módulo e a expressão final é dada por

$$\frac{C_i y[\phi(x_i)]}{\|w\|} \geq m, \quad i = 1, ..., n \tag{2.4}$$

Dada a expressão 2.4, o algoritmo consiste em encontrar o valor de $\|w\|$ tal que m seja máximo. Logo, o problema se reduz a um problema de otimização, em que a restrição é dada por $m\|w\| = 1$ para garantir a unicidade da solução. Uma vez estimado $\|w\|$, a classificação de uma nova observação é dada por

$$sinal[y(x_j)] = w^T \phi(x_j). \tag{2.5}$$

2.3 Algoritmos de Classificação Local via Particionamento

Esta seção apresenta mais detalhadamente algoritmos de classificação local que utilizam particionamento prévio e que foram utilizados como base para o desenvolvi-

mento deste trabalho. A subseção 2.3.1 descreve o método proposto por PERES e PEDREIRA [6], enquanto a subseção 2.3.2 descreve o algoritmo proposto por LEITE [16].

2.3.1 Abordagem Local-Global para Classificação

Em [6] foi descrito um novo método local-global de classificação que concilia abordagens supervisionadas e não-supervisionadas. Seu processo de classificação consiste em segmentar o conjunto de observações de treinamento em subconjuntos menores utilizando um algoritmo não-supervisionado para, então, classificar as observações de teste segundo uma regra local.

Seja um conjunto de observações de treinamento $\mathbf{x} = \{x_1, ..., x_n\}$ tal que $x_i \in \mathbb{R}^d$. A cada observação x_i está associada uma classe, podendo ser A ou B. Considere que $rótulo(x_i) = A$ caso x_i pertença à classe A ou $rótulo(x_i) = B$ caso x_i pertença à classe B. O método proposto por PERES e PEDREIRA [6] é composto por duas etapas: segmentação e classificação supervisionada local. Na primeira, o conjunto de observações $\mathbf{x}$ é segmentado utilizando um algoritmo não-supervisionado de quantização de vetores (vector quantization, VQ), que consiste em um processo iterativo de divisão do espaço em k regiões menores $\mathbf{R} = \{R_1, ..., R_k\}$ segundo um valor de k pré-determinado. Em seguida, a classificação de uma nova observação x_j é dada de acordo com a região R_i a qual ela pertence.

Suponha dois subconjuntos $r_i^A = \{x_i \in R_i | rótulo(x_i) = A\}$ e $r_i^B = \{x_i \in R_i | rótulo(x_i) = B\}$ e sejam as frequências de cada classe em R_i dadas por

$$f_i^A = \frac{\#r_i^A}{\#R_i} \quad \text{e} \quad f_i^B = \frac{\#r_i^B}{\#R_i} \tag{2.6}$$

respectivamente, em que # representa a cardinalidade do conjunto. Se $f_i^A = 1$ ou $f_i^B = 1$, a região R_i é dita homogênea e a classe de x_j pode ser dada por: $rótulo(x_j) = \{C | f_i^C = \max(f_i^A, f_i^B) \quad e \quad C \in \{A, B\}\}$, ou seja, a classe de x_j é correspondente à única classe presente em R_i. No entanto, caso $f_i^A < 1$ e $f_i^B < 1$, R_i é considerada heterogênea e a classificação de x_j é dada segundo um classificador bayesiano [8].

Considere a relação entre as frequências f_i^A e f_i^B dada por

$$\pi_k = \frac{f_i^A}{f_i^B} \tag{2.7}$$

e seja a relação de verossimilhança entre as classes em R_i, $L = p(x|A)/p(x|B)$, tal que

$$\hat{L}_x = \frac{[d(x, m_i^A)]^{-1}}{[d(x, m_i^B)]^{-1}} \tag{2.8}$$

em que $d(x, m_i^A)$ e $d(x, m_i^B)$ são as distâncias de x às médias das observações com classe A e B pertencentes a R_i respectivamente. O classificador bayesiano, então, classifica a observação x_j em uma região R_i heterogênea segundo a regra abaixo.

$$r\acute{o}tulo(x_j) = \begin{cases} A & se \quad \hat{L}_x \geq (\pi_k)^{-1} \\ B & caso \quad contrário \end{cases} \tag{2.9}$$

Em [6], o método acima descrito obteve resultados bem competitivos quando comparado a outros métodos de classificação muito utilizados, como o kNN e o SVM. Logo, particionamento seguido de classificação local demonstra ser uma via de métodos de classificação muito interessante.

2.3.2 Particionamento Dinâmico para Algoritmo Local de Classificação

Um método semelhante ao proposto por PERES e PEDREIRA [6] foi descrito em [16]. Diferentemente de PERES e PEDREIRA [6] que segmenta o conjunto de observações de treinamento uma única vez, LEITE [16] propôs um algoritmo que realiza um particionamento dinâmico e dispensa a pré-determinação da quantidade de regiões a serem encontradas nesse processo de segmentação. Trata-se de um método que também concilia abordagens supervisionadas e não-supervisionadas, mas que procura fornecer além da classe uma maior interpretabilidade local quanto ao resultado.

Seja um conjunto de observações de treinamento $\mathbf{x} = \{x_1, ..., x_n\}$ tal que $x_i \in \mathbb{R}^d$. A cada observação x_i está associada uma classe, podendo ser A ou B. Considere que $r\acute{o}tulo(x_i) = A$ caso x_i pertença à classe A ou $r\acute{o}tulo(x_i) = B$ caso x_i pertença à classe B. O método proposto por LEITE [16] consiste em segmentar iterativamente o conjunto $\mathbf{x}$ em regiões R_i utilizando o algoritmo de agrupamento k-médias. Suponha dois subconjuntos $r_i^A = \{x_i \in R_i | r\acute{o}tulo(x_i) = A\}$ e $r_i^B = \{x_i \in R_i | r\acute{o}tulo(x_i) = B\}$ e sejam as proporções de cada classe em R_i dadas por

$$P_i^A = \frac{\#r_i^A}{\#R_i} \quad \text{e} \quad P_i^B = \frac{\#r_i^B}{\#R_i} \tag{2.10}$$

em que # representa a cardinalidade do conjunto. A classificação de novas observações, então, é feita segundo as proporções das classes de cada região R_i encontrada.

O particionamento dinâmico visa estimar em quais e em quantas regiões o espaço de observações $\mathbf{x}$ deve ser dividido. Se resume a um processo em que o algoritmo de agrupamento k-médias é executado diversas vezes para crescentes quantidades de grupos k até que se atinja alguma das condições de parada propostas em [16]. Essas

condições são verificadas através de testes, os quais LEITE [16] nomeou como sendo Teste de Homogeneidade e Teste de Consistência.

A cada iteração do método, o valor de k é incrementado iterativamente até que se atinja alguma região homogênea (Teste de Homogeneidade) ou até que nenhuma das regiões seja consistente (Teste de Consistência). Uma região R_i, nesse caso, é dita homogênea quando $rótulo(x_i) = A$ para quase todo $x_i \in R_i$ ou $rótulo(x_i) = B$ para quase todo $x_i \in R_i$, e é dita heterogênea caso contrário. Além disso, considera-se consistente toda região R_i em que o desvio padrão de reamostragem das proporções P_i^* é inferior a um limiar estipulado.

Toda vez que o método chega a um valor k em que alguma região R_i é considerada homogênea, as observações $\mathbf{x} = \{x_i | x_i \in R_i\}$ são reservadas e o método reinicia a busca de um novo valor de k considerando apenas as observações restantes $\overline{\mathbf{x}} = \{x_i | x_i \notin R_i\}$. Esse processo é repetido até que $\overline{\mathbf{x}} = \emptyset$ ou até que todas as regiões R_i sejam inconsistentes. Ao final, o método retorna uma hierarquia de iterações, em que cada iteração m carrega um conjunto de k_m regiões $\mathbf{R}^m = \{R_1^m, ..., R_{k_m}^m\}$.

Assim, para classificar uma nova observação x_j, é preciso percorrer as iterações e verificar a que região homogênea ou heterogênea e inconsistente R_i^m pertence x_j. Uma vez que se sabe a região R_i^m, a classe de x_j é dada pela classe majoritária de R_i^m, ou seja, $rótulo(x_j) = \{C | P_i^C = \max(P_i^A, P_i^B) \quad e \quad C \in \{A, B\}\}$.

Capítulo 3

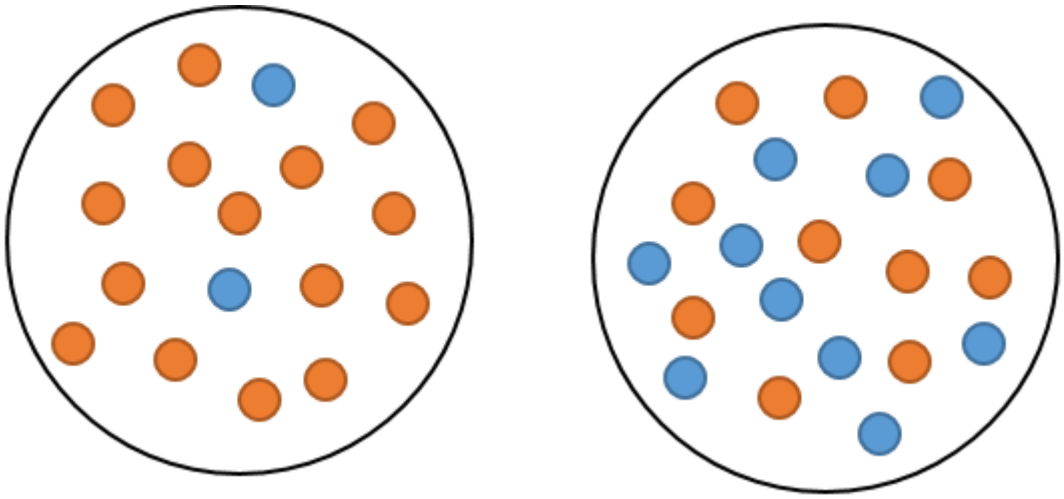
Metodologia

Esta dissertação propõe um novo método supervisionado que procura fornecer maior transparência e maior interpretabilidade quanto ao procedimento de classificação e aos resultados finais. O algoritmo consiste em um processo iterativo semelhante ao proposto por LEITE [16] que busca dividir o conjunto de observações de entrada em diversos subconjuntos segundo dois critérios a serem definidos no decorrer deste capítulo. A ideia central está calcada na possibilidade de se encontrar subconjuntos que contenham observações pertencentes à mesma classe, definindo regiões de fácil classificação.

Neste capítulo, detalha-se o algoritmo de classificação proposto, bem como os experimentos realizados durante o trabalho. Nas seções 3.1 e 3.2, são introduzidos alguns conceitos de caracterização de conjuntos e regiões, enquanto as seções 3.3, 3.4 e 3.5 descrevem o método proposto. A seção 3.6 apresenta as condições de implementação e o pseudocódigo do algoritmo. Por fim, a seção 3.7 descreve os conjuntos de dados artificiais e reais que foram utilizados nos experimentos propostos, cujos resultados se encontram no capítulo 4.

3.1 Regiões Homogêneas

Classificar facilmente significa classificar uma nova observação com alto grau de certeza. Intuitivamente, se um conjunto qualquer possui a grande maioria de seus elementos pertencentes à mesma classe, então todo novo elemento terá alta probabilidade de pertencer a essa classe. Diz-se, então, que esse conjunto é um conjunto homogêneo. Caso contrário, é considerado heterogêneo, conforme ilustrado na figura 3.1.



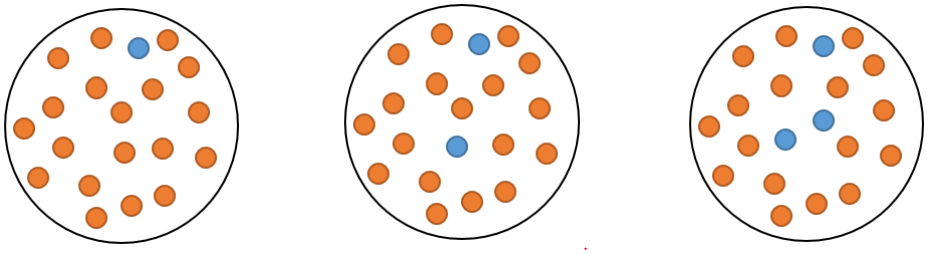
(a) Conjunto homogêneo (b) Conjunto heterogêneo

Figura 3.1: Ilustração de conjuntos homogêneos e heterogêneos

Assim, suponha que um conjunto $x = \{x_1, ..., x_n\}$ de n observações seja dividido em vários subconjuntos, formando distintas regiões no espaço de atributos. Cada região, caracterizada por um subconjunto de observações, pode ser homogênea ou heterogênea conforme a definição 3.1.1.

Definição 3.1.1. Uma região R_i é dita homogênea se $(100 - p)\%$ das observações $x_j \in Ri$ possuem o mesmo rótulo de classe para um pequeno valor de p (e.g. $p < 10$). Caso contrário, ela é dita heterogênea.

Assim, uma região R_i é considerada homogênea se todas, ou quase todas, as observações que a compõem pertencem à mesma classe, A ou B. Regiões homogêneas, portanto, configuram subproblemas em que classificar uma nova observação não exige muito esforço. São regiões que, ao longo do treinamento do algoritmo, devem ser preservadas e protegidas. Definir se uma região R_i é homogênea ou não depende, então, do parâmetro p, cujo valor vai depender da tolerância a erros permitida pela aplicação. A figura 3.2 ilustra a influência do valor de p na configuração da região.



(a) $p = 5\%$ (b) $p = 10\%$ (c) $p = 15\%$

Figura 3.2: Ilustração de conjuntos homogêneos e tolerância de homogeneidade.

3.2 Regiões Separáveis

Uma região separável é aquela que admite vantagens em ser dividida. Se um conjunto de n observações $x = \{x_1, ..., x_n\}$ é heterogêneo, mas contém uma subrregião homogênea, então ele é passível de ser dividido. A figura 3.3 apresenta um conjunto heterogêneo em que uma subrregião homogênea poderia ser considerada.

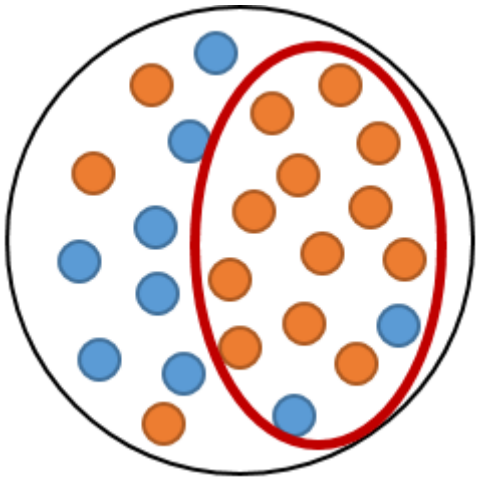


Figura 3.3: Conjunto heterogêneo com subrregião homogênea considerando $p = 15\%$

A capacidade de uma região admitir subrregiões homogêneas, como ilustrado na figura 3.3, está diretamente associada à diferença entre as distribuições de cada classe e à disposição espacial de cada observação. A figura 3.4 ilustra a diminuição da sobreposição de classes à medida em que se afasta as médias de cada distribuição. Nesse caso, quanto mais distantes são as distribuições, maior é a chance de encontrar subrregiões homogêneas.

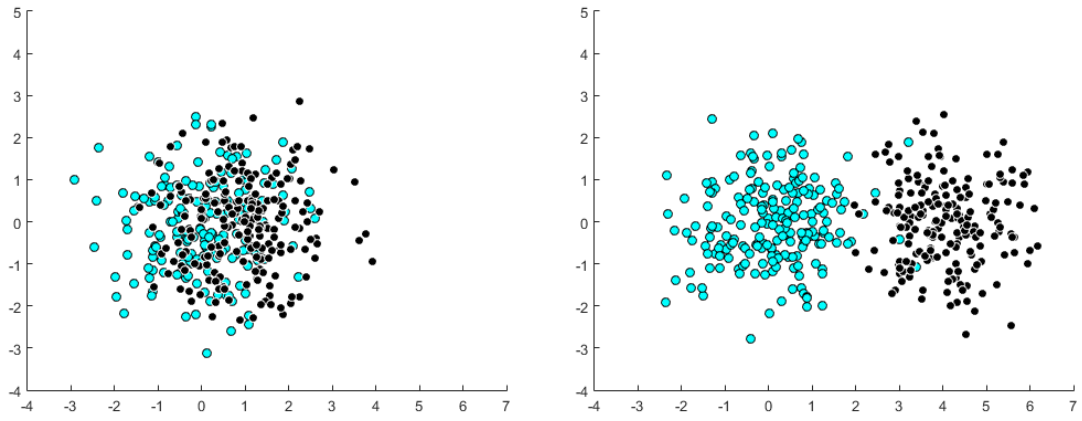


Figura 3.4: Diferença de sobreposição dada a distribuição espacial das classes.

Medir a distância entre as distribuições e, consequentemente a sobreposição das classes, configura um problema de divergência entre duas funções de densidade de probabilidade [5]. Trata-se de uma pseudo distância que retrata o quanto uma distribuição difere de outra. Diversas abordagens para calcular o grau de divergência entre duas distribuições são propostas na literatura [17, 18]. Dentre as principais, uma das mais usadas é a divergência de Kullback-Leibler [5, 17]. Entretanto, essa abordagem não resulta em uma distância simétrica e não permite forma fechada para diversos tipos de distribuição, como por exemplo mistura de gaussianas [18].

A simetria aqui é importante para que não seja necessário estipular um sentido de cálculo, permitindo maior consistência na análise da separabilidade do conjunto em diversas instâncias. A divergência entre as distribuições deve retratar o mesmo cenário independente de qual delas seja escolhida como referência. Caso contrário, decidir a separabilidade do conjunto estaria sujeito a uma instabilidade imprevisível, pois a partir de uma classe o conjunto poderia ser separável, mas a partir de outra poderia não ser.

Ademais, a desvantagem da Kullback-Leibler em não permitir forma fechada e analítica em diversos casos torna difícil o cálculo da divergência e faz com que seu resultado seja dependente do número de iterações utilizadas durante o cálculo. Nesse sentido, o ideal é que a divergência entre as distribuições seja simétrica e possa ser calculada analiticamente. Trabalhos recentes [18, 20] analisaram, em diversos casos, o desempenho da divergência de Cauchy-Schwarz [19, 20, 23], que além de simétrica pode ser estimada analiticamente. Assim, esta dissertação propõe que a separabilidade de um conjunto de observações seja definida a partir da divergência de Cauchy-Schwarz $D_{r,q}$ entre r e q definida como segue:

$$D_{r,q} = -\log \frac{\left(\int r(x)q(x)\mathrm{d}x\right)^2}{\int r^2(x)\mathrm{d}x \int q^2(x)\mathrm{d}x}. \tag{3.1}$$

Em uma dada região R_i de observações, seja Q_A a função de densidade de probabilidade estimada (PDF) de $\{x \in R_i, rótulo(x) = A\}$ e Q_B a PDF estimada de $\{x \in R_i, rótulo(x) = B\}$, podemos definir a divergência entre as classes como sendo $D_{A,B}$. Assim, podemos dizer que a partir de um dado valor de $D_{A,B}$, as distribuições se encontram suficientemente divergentes, ou seja, há na região R_i algum subconjunto que poderia ser considerado homogêneo. Nesse caso, dividir R_i permite alcançá-lo. Caso contrário, as distribuições se encontram fortemente sobrepostas e subdividir a região levaria muito provavelmente a um sobreajuste.

Definição 3.2.1. Uma região R_i é dita separável se $D_{A,B}$ entre Q_A e Q_B é maior ou igual a um limiar previamente estipulado. (e.g. $s = 0.5$).

A estimação de densidade de probabilidade pode ser realizada aplicando métodos

de estimação por kernels, funções não-negativas cuja integral vale um [8], que representam distribuições locais e funcionam como janelas de Parzen [8, 17]. Assim, a estimativa da função de densidade pode ser escrita matematicamente conforme abaixo.

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K_h \left(\frac{x - x_i}{h} \right), \tag{3.2}$$

onde K_h representa a função kernel a ser utilizada como estimativa da distribuição local e h representa a largura da janela. Por simplicidade, em geral são utilizados kernels normais ou gaussianos com média μ e variância σ^2, que podem ser definidos como abaixo [17, 20].

$$G_{\mu,\sigma^2} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{|x - \mu|^2}{2\sigma^2} \right). \tag{3.3}$$

Para aplicações multidimensionais, isto é, com $d > 1$ dimensões, esse kernel gaussiano passa a ser uma função multivariada em que a variância dá lugar à matriz de covariância H.

$$K_H(x) = \frac{1}{\sqrt{(2\pi)^d |H|}} \exp\left(-\frac{1}{\sqrt{x^T H^{-1} x}} \right). \tag{3.4}$$

No caso, tanto a matriz de covariância H quanto a variância σ^2 definem a largura das gaussianas a serem utilizadas na estimativa da função de densidade, logo são parâmetros que devem ser devidamente ajustados. Nesse sentido, há diversos trabalhos que propõem maneiras de chegar a um valor ótimo desses parâmetros. SILVERMAN [21] propõe uma formulação matemática que permite considerar um valor ótimo estimado para o parâmetro σ^2 da expressão 3.3 em aplicações unidimensionais, o que ficou conhecido como a regra de Silverman (Silverman's Rule of Thumb). SCOTT [22], por outro lado, expandiu esse método para casos multidimensionais, o que permite considerar a matriz de covariância H como sendo:

$$\hat{H} = \mathbf{I}\sigma \left(\frac{4}{(d+2)n} \right)^{(d+4)^{-1}}, \tag{3.5}$$

onde d é o número de dimensões da aplicação, I representa uma matriz identidade $d \times d$, σ é o desvio padrão dos dados da amostra e n representa o tamanho da amostra cuja função de densidade de probabilidade se está estimando. Essa expressão permite a estimação da PDF de forma não-paramétrica, reduzindo o impacto da escolha errada de parâmetros.

As distribuições Q_A e Q_B podem ser estimadas através de métodos que se utilizam de kernels gaussianos, conforme a expressão 3.4. Seja n_A e n_B o número de observações que pertencem à classe A e à classe B respectivamente. Nesse caso,

temos que:

$$\begin{aligned} Q_A &= \frac{1}{n_A}\sum_{i=1}^{n_A} K_{H_A}(x - x_i) \quad x_i \in \mathbf{x} \quad e \quad rótulo(x_i) = A \\ Q_B &= \frac{1}{n_B}\sum_{i=1}^{n_B} K_{H_B}(x - x_i) \quad x_j \in \mathbf{x} \quad e \quad rótulo(x_j) = B \end{aligned} \tag{3.6}$$

A matriz de covariância H aqui é estimada de acordo com a regra de SCOTT [22] apresentada no Capítulo 2, seção 2.1. Considerando A e B na equação de $D_{r,q}$, podemos estimar os termos como a seguir [20, 23]:

$$\begin{aligned} \int r(x)q(x)\mathrm{d}x &= \frac{1}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\int K_{H_A}(x - x_i)K_{H_B}(x - x_j)\mathrm{d}x \\ &= \frac{1}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} K_{H_{AB}}(x_i - x_j), \end{aligned} \tag{3.7}$$

$$\begin{aligned} \int r^2(x)\mathrm{d}x &= \frac{1}{n_A^2}\sum_{i=1}^{n_A}\sum_{j=1}^{n_A} K_{H_A}(x_i - x_j), \\ \int q^2(x)\mathrm{d}x &= \frac{1}{n_B^2}\sum_{i=1}^{n_B}\sum_{j=1}^{n_B} K_{H_B}(x_i - x_j), \end{aligned} \tag{3.8}$$

onde H_{AB}, H_A e H_B são definidas de considerando a regra de SCOTT [22] e as adaptações necessárias às estimativas acima.

$$\begin{aligned} H_A &= 2\mathbf{I}\left[\sigma_A\left(\frac{4}{(d+2)n_A}\right)^{(d+4)^{-1}}\right]^2, \\ H_B &= 2\mathbf{I}\left[\sigma_B\left(\frac{4}{(d+2)n_B}\right)^{(d+4)^{-1}}\right]^2, \\ H_{AB} &= \mathbf{I}\left\{\left[\sigma_A\left(\frac{4}{(d+2)n_A}\right)^{(d+4)^{-1}}\right]^2 + \left[\sigma_B\left(\frac{4}{(d+2)n_B}\right)^{(d+4)^{-1}}\right]^2\right\}. \end{aligned} \tag{3.9}$$

Considerando as aproximações acima e as propriedades de logaritmo, $D_{A,B}$ pode ser reduzida a uma expressão que depende apenas das observações e seus rótulos de classe [19, 20].

$$
\begin{aligned}
D_{A,B} = \frac{1}{n_A^2} \sum_{i=1}^{n_A} \sum_{j=1}^{n_A} K_{H_A}(x_i - x_j) + \\
+ \frac{1}{n_B^2} \sum_{i=1}^{n_B} \sum_{j=1}^{n_B} K_{H_B}(x_i - x_j) - \\
- \frac{2}{n_A n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} K_{H_{AB}}(x_i - x_j).
\end{aligned}
\tag{3.10}
$$

Em suma, a expressão 3.10 retorna um valor inversamente proporcional à sobreposição das classes. Quanto mais sobrepostas, menor é o valor de $D_{A,B}$, então quanto maior o valor de $D_{A,B}$, maior a probabilidade de haver uma subrregião homogênea.

3.3 Algoritmo de Classificação

O método proposto neste trabalho utiliza os conceitos de regiões homogêneas e de regiões separáveis para dividir o espaço das observações em regiões de fácil e de difícil classificação. Assim, seja $x = \{x_1, ..., x_n\}$ um conjunto de n observações de dimensão d às quais são atribuídos rótulos de classe, A ou B. As dimensões d representam os atributos de cada dado de entrada, cuja interpretação depende da aplicação em questão. Suponha que esse conjunto seja particionado em k subconjuntos, de maneira que cada subconjunto i representa uma região R_i centrada em um protótipo V_i. Nesse sentido, cada região pode ser vista, também, como um subconjunto de $\mathbb{R}^d$ e a união dessas regiões representada pelo conjunto $L = \{V_1, ..., V_i, ..., V_k\}$ de protótipos formam o espaço $\mathbb{R}^d$. Considera-se que uma observação $x_j \in R_i$ tem uma probabilidade P_i de ter rótulo A e $(1 - P_i)$ de ter rótulo B, de forma que cada região R_i tenha uma dada probabilidade de ser classificada como A ou como B. Assim, qualquer nova observação que caia na região R_i receberá o rótulo de acordo com a classe mais provável e frequente dessa região.

Para dividir o espaço em regiões menores, é necessária a execução de um algoritmo de agrupamento. Nesta dissertação, esse agrupamento foi feito utilizando o algoritmo k-médias, assim como em [16]. Para que não haja a necessidade de se pré-definir o valor de k, a divisão do espaço é feita de maneira iterativa e incremental. Ou seja, começa-se com $k = 1$ e incrementa-se esse valor reexecutando o k-médias a cada iteração até que um dado critério de parada seja alcançado. Esse processo é interrompido toda vez que se alcança alguma região R_i em que todas, ou quase todas, as observações pertençam à mesma classe. Nesse ponto, tem-se uma ou mais regiões de fácil classificação que devem ser mantidas, logo elas devem ser retiradas do processo de divisão e este deve ser reiniciado com $k = 1$. Essa sequência

é executada até que não haja mais observações no conjunto a ser dividido ou até que não seja mais possível alcançar essas regiões homogêneas.

Os critérios que definem os passos do processo de divisão estão relacionados às duas principais etapas do algoritmo: teste de homogeneidade e teste de separabilidade. O primeiro verifica se na iteração corrente há alguma região homogênea, já o segundo verifica se as regiões heterogêneas são separáveis, conforme apresentado nas seções 3.1 e 3.2. Assim, a cada iteração em que se divide o espaço, o método testa a presença de regiões homogêneas e, em caso negativo, verifica se há alguma região separável a fim de continuar dividindo.

A figura 3.5 apresenta o fluxograma do algoritmo, em que Figura 3.5(c) corresponde ao teste de homogeneidade e Figura 3.5(g), Figura 3.5(h) e Figura 3.5(i) correspondem ao teste de separabilidade. Conforme sinalizado pelo fluxograma e melhor descrito na seção 3.6, o passo Figura 3.5(d) garante que as regiões de fácil classificação são mantidas intactas durante o treinamento do método.

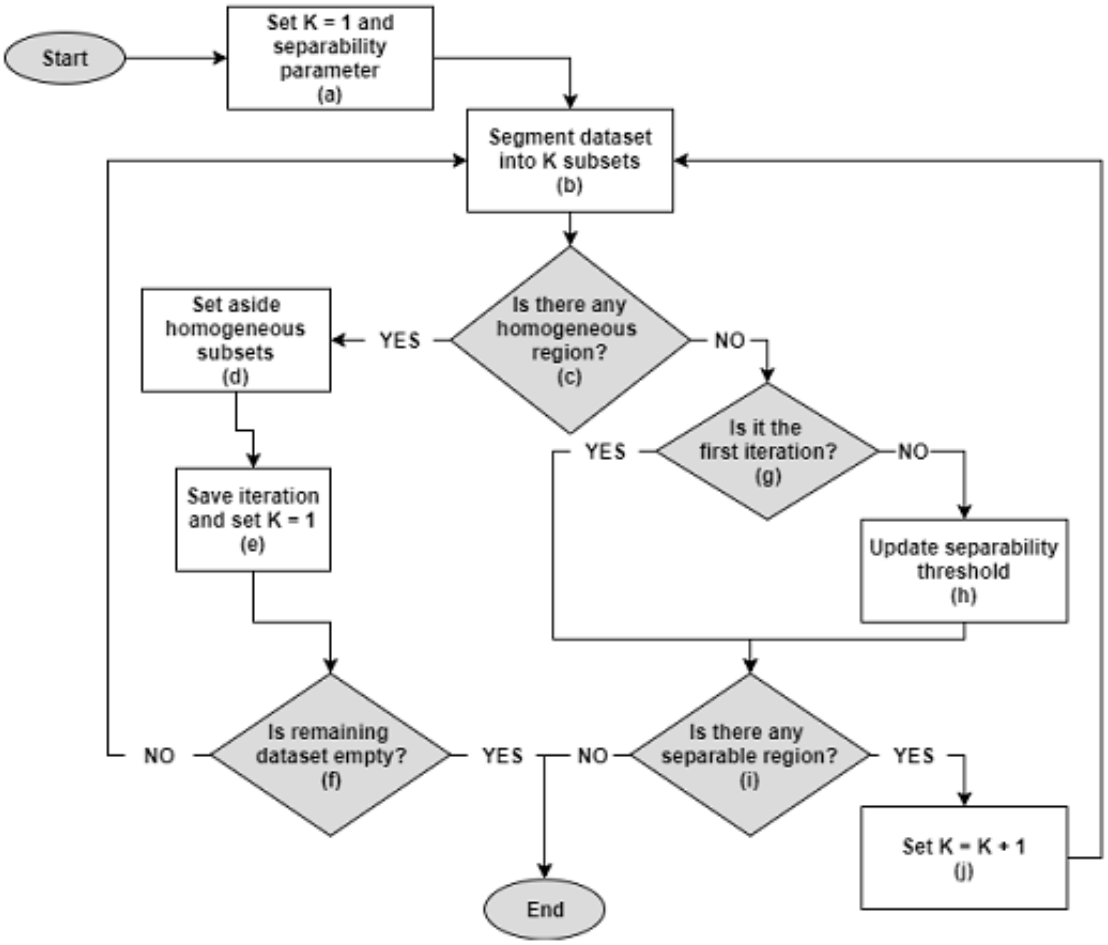


Figura 3.5: Fluxograma - Algoritmo de Classificação por Particionamento Hierárquico (ACPH)

O teste de homogeneidade permite decidir se o processo de divisão deve retornar a $k = 1$ ou deve prosseguir incrementando k. Caso o algoritmo alcance uma região homogênea, esse deve realizar um novo processo de fragmentação considerando apenas as regiões que não são tidas como homogêneas segundo a Definição 3.1.1. A Figura 3.6, em que a classe A corresponde aos pontos pretos e a classe B corresponde aos pontos azuis, mostra a sequência de iterações até que o método alcance uma região homogênea com, no caso, 0% de classe A conforme Figura 3.6(e). O gráfico Figura 3.6(f) mostra o conjunto de dados que restou ao reservar os dados referentes à região homogênea.

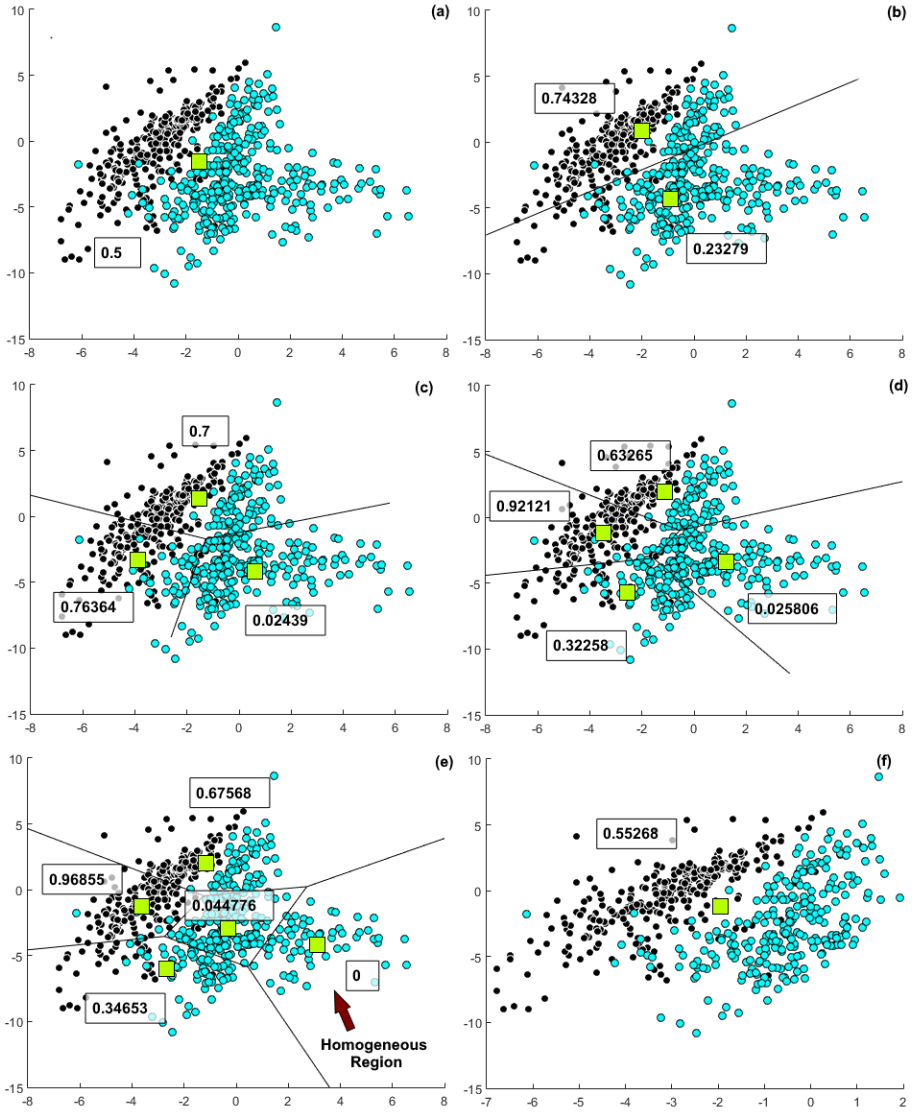


Figura 3.6: ACPH segue dividindo o espaço até que alcance uma ou mais regiões homogêneas. De (e) para (f), ACPH reserva a região homogênea. Os números nos retângulos são os valores de proporção da classe A, representada por marcadores pretos. Os quadrados verdes são os protótipos que representam as regiões.

3.4 Grau de Localidade

Em LEITE [16], a medida de localidade é dada pelo desvio padrão de reamostragem da proporção local das classes, mas para os diversos experimentos realizados durante este trabalho, a reamostragem intra-regional resultou em um conjunto de operações relativamente custosas, porém com pouco ganho informacional.

Quando o método se torna muito local, passa a tratar de poucas observações, o que faz com que medidas estatísticas calculadas a partir delas sejam pouco confiáveis. A técnica de bootstrapping consiste em uma reamostragem com reposição visando criar diversas versões da amostra disponível [8, 24]. Assim, suponha que a amos-

tra tenha n observações e que sejam feitas M reamostragens de tamanho n com reposição, logo cada nova amostra provavelmente terá pontos duplicados. Uma medida estatística θ, nesse caso, é dada pela média entre os valores de θ_b calculados para cada nova amostra b, conforme a expressão abaixo.

$$\hat{\theta} = \frac{1}{B}\sum_{b=1}^{B}\theta_b. \tag{3.11}$$

A estimativa da variância é dada pela média dos quadrados das diferenças entre cada valor θ_b e a medida estimada $\hat{\theta}$.

$$Var_{boot}\left(\hat{\theta}\right) = \frac{1}{B}\sum_{b=1}^{B}\left(\theta_b - \hat{\theta}\right)^2 \tag{3.12}$$

é possível perceber que à medida em que B tende a infinito, a variância Var_{boot} tende a zero, logo quanto maior a quantidade de reamostragens B, mais confiante é a estimativa da estatística $\hat{\theta}$.

A Tabela 3.1 demonstra que, ainda que a proporção seja calculada sem nenhuma reamostragem, ela ainda representa bem as regiões para um dos experimentos realizados, uma vez que o desvio padrão de reamostragem é significativamente pequeno. Resultados semelhantes foram obtidos em diversos outros experimentos.

Tabela 3.1: Reamostragem (*bootstrap*) para teste de consistência da proporção de classe local. Não há a necessidade de utilizar técnicas de reamostragem para estimar o valor de proporção.

Teste de Consistência para Proporção Local

Proporção da Classe A (%)	Reamostragem *bootstrap*	
	Média da Proporção (%)	Desvio Padrão da Proporção (%)
13.3	13.2	5.6
32.3	30.4	7.5
35.1	33.8	6.3
51.5	50.2	8.4
93.1	93.1	5.3

Dessa forma, esta dissertação propõe que o valor da divergência calculado para cada região no teste de separabilidade seja interpretado como um indicativo de localidade. Caso as distribuições estejam muito sobrepostas e, portanto, a divergência assuma um valor muito baixo, não há motivos para continuar dividindo e, nesse caso, conclui-se que a região em questão é a mais local possível e por isso não pode ser considerada separável. Nesse sentido, o limiar que define a partir de que valor uma região R_i é considerada separável deve considerar que as regiões se tornam cada

vez mais locais. A atualização desse limiar ao longo do treinamento é realizada a partir da variação do erro de classificação entre as duas últimas iterações conforme a expressão abaixo:

$$(limiarD_{A,B})_i = \frac{(limiarD_{A,B})_{i-1}}{\sqrt{(taxadeerro)_i/(taxadeerro)_{i-1}}}. \tag{3.13}$$

A expressão 3.13 considera que o limite de divergência tolerável deve ser aumentado quando a taxa de erro atual for inferior à anterior. Essa proporcionalidade inversa permite que se exija distribuições cada vez mais divergentes à medida em que o algoritmo se torna cada vez mais local ou à medida em que ele se distancia da classificação ótima. A raíz quadrada no denominador faz com que o impacto da variação da taxa de erro sobre o limiar seja menor e evita que uma iteração discrepante se sobreponha sobre as demais iterações.

A atualização do limiar, portanto, exige que o classificador intermediário de cada iteração seja utilizado para classificação toda vez que o teste de separabilidade é executado. Essa dinamicidade do valor de separabilidade minimiza a formação de regiões muito locais e, consequentemente, o sobreajuste do método.

3.5 Condições de Parada e Classificador Final

O treinamento do método se encerra quando não é mais possível alcançar regiões homogêneas. Esse cenário ocorre quando não há mais observações em regiões heterogêneas ou quando todas as regiões heterogêneas são inseparáveis, ou seja, $D_{A,B}$ é inferior ao limiar de separabilidade para toda região R_i participante do processo de divisão. No último caso, as regiões heterogêneas e inseparáveis descrevem a porção do espaço em que existe uma incerteza significativa quanto à classificação. Nesse momento, o algoritmo encerra as iterações e retorna o modelo final de classificação. A Figura 3.7 demonstra um exemplo em que o algoritmo se depara com apenas regiões heterogêneas e inseparáveis.

Uma vez concluído o processo de treinamento, a saída do método consiste em uma estrutura que armazena uma hierarquia de iterações em que regiões homogêneas foram encontradas. Caso o algoritmo tenha sido encerrado em uma iteração com apenas regiões heterogêneas e inseparáveis, essa última iteração é incorporada à hierarquia de saída. Assim, pode-se considerar que a cada nível da hierarquia, há sempre alguma região na qual a classificação já é possível de ser realizada.

Suponha uma observação de teste x_i, para dizer se essa observação pertence à classe A ou à classe B, é necessário verificar a que região ela pertence. O algoritmo de predição percorre as iterações salvas na hierarquia de saída verificando a pertinência de x_i às regiões encontradas durante o treinamento, até que a observação caia em

uma região homogênea ou até que a última iteração seja alcançada. Dessa forma, se em uma iteração j a observação x_i pertencer a uma região homogênea, então a classe de x_i será dada pela classe majoritária dessa região. Caso contrário, se x_i pertencer a uma classe heterogênea e a iteração j não for a última, então o classificador passa a verificar a pertinência de x_i às regiões da iteração $(j+1)$. Se o algoritmo alcançar a última iteração e se x_i pertencer a uma região heterogênea dessa iteração, então o classificador atribui à observação a classe majoritária e retorna a confiança desse resultado baseado nas proporções locais das classes.

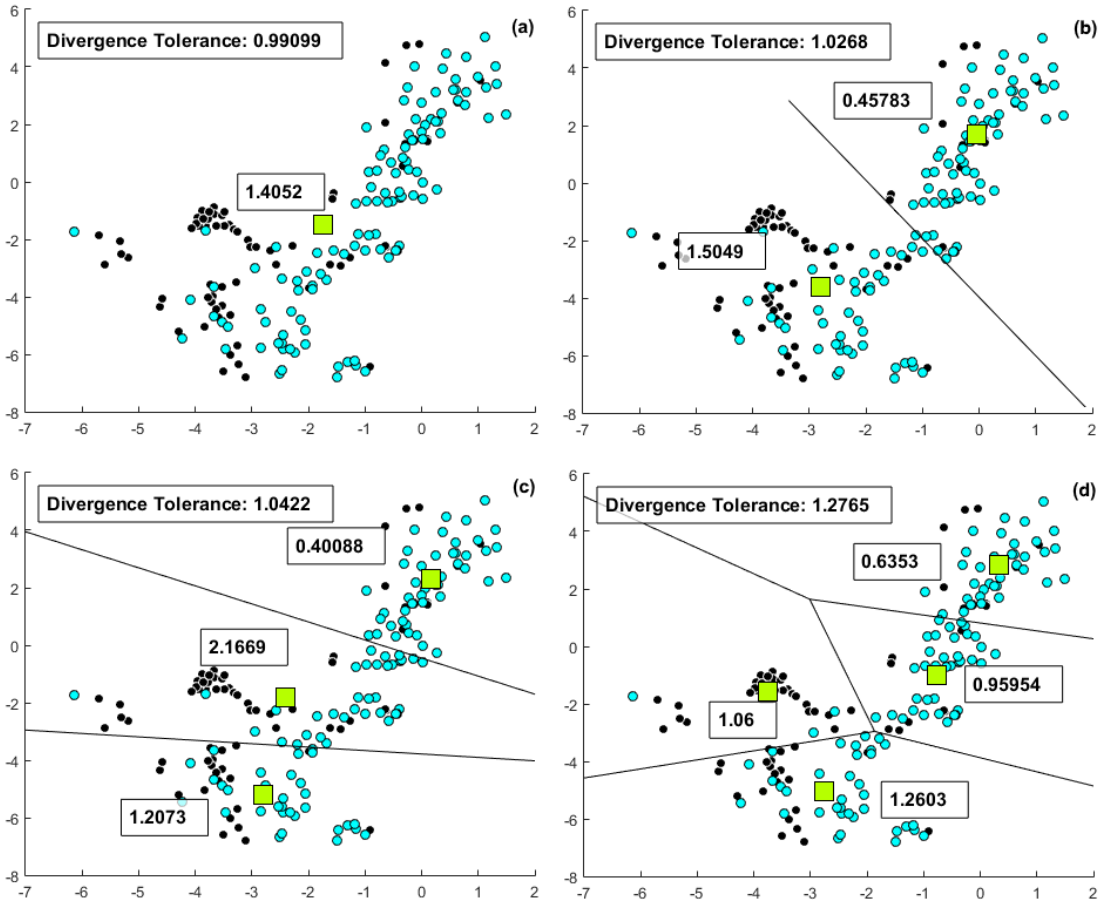


Figura 3.7: Critério de parada baseado na divergência de Cauchy-Schwarz. Os números nos retângulos representam a divergência de Cauchy-Schwarz de cada região. Se todas as divergências forem inferior ao limiar, então o algoritmo deve ser interrompido.

3.6 Implementação do Algoritmo

A implementação do algoritmo proposto nesta dissertação foi realizada em ambiente de MatLab, utilizando, em geral, funções e parâmetros padrões. O pseudocódigo 1 do algoritmo resume suas etapas. Aqui, o histórico de treinamento $\mathcal{H}$ consiste em uma

estrutura de dados a ser utilizada para armazenamento das iterações em que regiões homogêneas foram encontradas ou iterações em que nenhuma região é separável.

Data: Set of Observations $\mathbf{x} = \{x_1, ..., xn\}$ and Class Labels $\mathcal{C} \in A, B$
Result: Nested Set of Final Prototypes $L = \{V_1, ..., V_k\}$ that represent Homogeneous Regions and Training History $\mathcal{H}$

Let *cauchy_schwarz_divergence*$(*)$ be the function that calculates the Cauchy-Schwarz Divergence;
Set homogeneity parameter as ***h_param***;
Set initial Cauchy-Schwarz threshold as ***cs_threshold***;
Set $\mathbf{k} = 1$;

while ***x*** *not empty* **do**
  Let j refers to current iteration;
  Let V_j be the set of prototypes that represent homogeneous regions;
  Divide $\mathbf{x}$ into $\mathbf{k}$ subsets using *k-means*;
  Add the $\mathbf{k}$ prototypes to $\mathcal{H}^j_{prototype}$;
  foreach *subset* S_i, $i \in [1, k]$ **do**
    n_A = number of observations in $\{x \in S_i \ / \ \mathcal{C}(x) == 1\}$;
    n_B = number of observations in $\{x \in S_i \ / \ \mathcal{C}(x) == 2\}$;
    Add $n_A/(n_A + n_B)$ to $\mathcal{H}^j_{proportion}$;
    if $n_A/(n_A + n_B) <$ ***h_param*** *or* $n_B/(n_A + n_B) <$ ***h_param*** **then**
      Add subset prototype to V_j;
      Remove subset $x \in S_i$ from $\mathbf{x}$;
    end
  end
  Add $\mathcal{H}^j_{proportion}$ and $\mathcal{H}^j_{prototype}$ to $\mathcal{H}$;
  if V_j *not empty* **then**
    Set $\mathbf{k} = 1$;
    Add V_j to L;
  else
    Set V_j to empty;
    foreach *subset* S_i, $i \in [1, k]$ **do**
      cs_divergence = *cauchy_schwarz_divergence*(S_i);
      if *cs_divergence* $>$ ***cs_threshold*** **then**
        $\mathbf{k} = \mathbf{k} + 1$;
        separable = true;
      else
        Add subset prototype to V_j;
      end
    end
    if *not separable* **then**
      Set $\mathbf{x}$ to empty;
      Add V_j to L;
    end
  end
end

Algoritmo 1: Pseudocódigo de Algoritmo de Classificação via Particionamento Hierárquico (ACPH)

3.7 Bancos de Dados e Experimentos

Para avaliar o desempenho do método de classificação por particionamento hierárquico, diversos experimentos foram realizados utilizando bancos de dados artificiais e bancos de dados reais obtidos no repositório UCI [1]. Os bancos sintéticos, conforme a Figura 3.8, foram produzidos utilizando mistura de gaussianas com média e desvio padrão aleatórios.

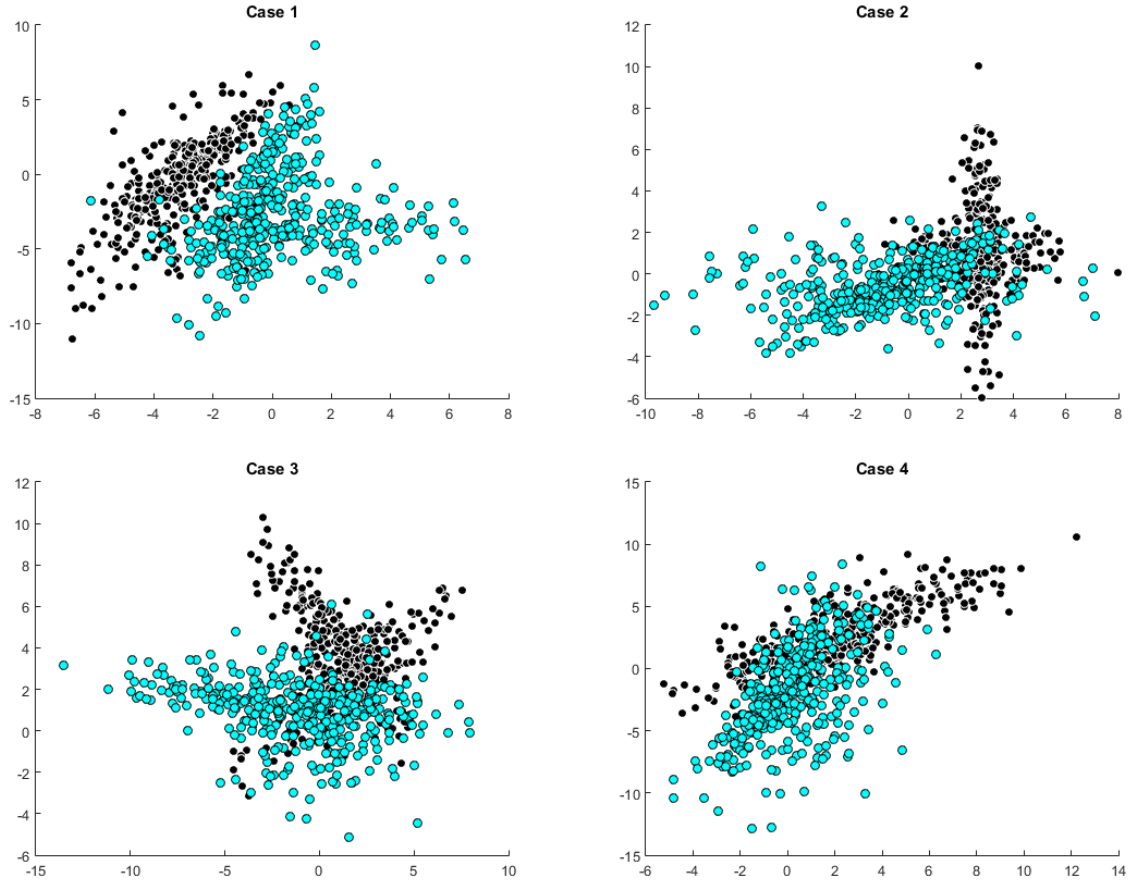


Figura 3.8: Representação visual dos casos sintéticos que serão utilizados nos experimentos descritos nos Capítulos 3 e 4. Aqui, a classe A é dada pelos pontos pretos.

A respeito dos bancos de dados do UCI, três são originalmente binários, mas os demais foram pré-processados de forma a manter apenas duas classes. Os dois bancos de Câncer de Mama, bem como o de Hipertireoidismo possuem classes referentes à presença ou ausência da doença. Os conjuntos de dados referentes ao Uso de Métodos Contraceptivos, à Avaliação de Carros e à Necessidade de Creche tinham mais de duas classes, mas todos os três apresentavam uma classe muito mais populosa que as demais. Assim, para esses casos, a classe A representa a classe majoritária e a classe B representa a união das demais classes minoritárias. Ademais, esses três últimos

[1] https://archive.ics.uci.edu/ml/index.php

bancos possuem atributos discretos referentes a características físicas de pessoas e objetos, além de atributos pessoais como idade.

Tanto os dados de Diagnóstico quanto os dados de Prognóstico de Câncer de Mama foram extraídos de imagens digitais tiradas de massas presentes nos seios. No caso da aplicação de prognóstico, os dados são dados de acompanhamento e o objetivo é prever a recorrência do câncer. Todos os três bancos referentes a doenças possuem atributos contínuos. Assim, é possível comparar o desempenho do algoritmo quando aplicados a casos discretos e a casos contínuos. Mais detalhes a respeito dos bancos do UCI podem ser encontrados na tabela 3.2.

Tabela 3.2: Descrição e distribuição das classes dos bancos de dados do repositório do UCI.

Bancos de Dados do UCI

Bancos	Atributos	Instâncias	Classificação Positiva (classe A)		Classificação Negativa (classe B)	
			Descrição	N. Obs.	Descrição	N. Obs.
Câncer de Mama Diagnóstico	32	569	Tumor Maligno	212	Tumor Benigno	357
Câncer de Mama Prognóstico	32	198	Câncer Recorrente	47	Sem Recorrência	151
Doença na Tireoide	5	215	Hipertireoidismo	35	Sem Hipertireoidismo	180
Uso de Métodos Contraceptivos	9	1473	Usa Métodos Contraceptivos	844	Não Usa Métodos Contraceptivos	629
Avaliação de Carros	6	1728	Carro é pelo menos aceitável	518	Carro inaceitável	1210
Creche	8	12960	Recomendado	8640	Não Recomendado	4320

3.7.1 Estimativa Fora da Amostra

Avaliar métodos de classificação exige a análise do desempenho em conjuntos de dados que não foram utilizados no treinamento do algoritmo, logo é preciso ter um subconjunto de treinamento e um subconjunto de teste. Em diversas aplicações, no entanto, é inviável a divisão da amostra, uma vez que resultaria em um subconjunto de treinamento muito reduzido, comprometendo a capacidade de generalização do modelo preditivo. Nesses casos, é aconselhável o uso de técnicas como a validação cruzada (cross-validation) [7], em que o algoritmo de classificação é treinado e testado múltiplas vezes em diferentes subconjuntos disjuntos da amostra. A seguir, são descritas duas dessas técnicas:

- **k-*Folds*:** Esse método consiste em dividir o conjunto de dados de entrada em k subconjuntos disjuntos a serem utilizados em k iterações de classificação.

Cada subconjunto é utilizado como conjunto de teste, enquanto os demais formam o conjunto de treinamento. Assim, ao final das k iterações, todos os subconjuntos terão sido usados como teste e a estimativa do erro preditivo do algoritmo de classificação é dada pela média entre os erros preditivos obtidos. Esse processo permite melhor avaliação da generalização do algoritmo.

- ***Leave-One-Out*:** Quando k-folds é executado em um conjunto de n observações utilizando $k = n$, temos uma validação cruzada em que a cada iteração, uma observação é considerada teste e as $(k - 1)$ observações restantes formam o conjunto de treinamento. Da mesma forma que o k-folds, o erro preditivo do algoritmo de classificação é estimado de acordo com a média entre os erros preditivos das iterações. Nesse caso, a média dos erros será equivalente à quantidade de observações que foram classificadas equivocadamente.

Os experimentos realizados se diferem quanto à estimativa do desempenho. Para os bancos sintéticos, os dados de entrada, disponíveis em qualquer quantidade, foram separados em grupo de treinamento e grupo de teste. Porém, para os bancos do UCI, foram utilizadas técnicas de validação cruzada. Para aqueles com poucas observações disponíveis, como os conjuntos de câncer de mama e o de hipertireoidismo, a validação cruzada desempenhada foi do tipo leave-one-out, enquanto para os demais bancos do UCI, foi utilizado k-fold, já que esses tinham significativamente mais observações.

A ideia central dos experimentos foi avaliar o desempenho do Algoritmo de Classificação por Particionamento Hierárquico (ACPH), descrito nesta dissertação, frente a outros algoritmos comumente usados: Máquina de Vetores de Suporte (support vector machine, SVM) e k-vizinhos mais próximos (k-nearest neighbors, kNN.). Comparar apenas o desempenho global, nesse caso, é trair o método proposto, já que esse realiza uma classificação independente para cada região encontrada. Nesse sentido, todas as comparações foram realizadas considerando as regiões finais do ACPH e, portanto, podem ser divididas em três casos: comparação global, comparação para observações que pertencem a regiões homogêneas do ACPH e comparação para observações que pertencem a regiões heterogêneas do ACPH.

Como o propósito aqui não é estressar os métodos de classificação, muito menos executá-los em situações adversas, tanto kNN quanto SVM foram executados a partir de parâmetros padrões, com exceção do número de vizinhos k, para o qual não há valor padrão. Ainda assim, existe uma heurística bem aceita na área de reconhecimento de padrões que propõe que o valor de k seja dado pela raiz quadrada do número de observações do conjunto de treinamento [8]: $k = \sqrt{NúmerodeObservaçõesdeEntrada}$.

Capítulo 4

Resultados e Discussões

Este capítulo tem por finalidade apresentar e discutir os resultados obtidos com os experimentos descritos no capítulo 3. Aqui os resultados são divididos de tal forma que a seção 4.1 apresenta a comparação de desempenho de classificação entre os algoritmos e a seção 4.2 discute o ganho de informação e interpretabilidade que os algoritmos kNN e SVM ganham quando analisados sob o ponto de vista do algoritmo proposto ACPH.

4.1 Desempenho Comparado de Classificação por Particionamento Hierárquico

Em primeiro lugar, é preciso analisar a competitividade do método proposto frente a outros métodos comumente usados. As tabelas 4.1 e 4.2 mostram os resultados obtidos ao executar o algoritmo proposto e os algoritmos kNN e SVM para os bancos descritos no capítulo 3. é possível perceber que, considerando os resultados globais, regiões com alta confiança são, normalmente, mascaradas devido às regiões com baixa confiança. Uma melhor análise, portanto, deve considerar regiões homogêneas e regiões heterogêneas separadamente. Para isso, a classificação fora da amostra para os algoritmos kNN e SVM considera as regiões encontradas pelo algoritmo ACPH, ou seja, foi estimado o desempenho para observações que, segundo o ACPH, pertencem a regiões homogêneas e, em paralelo, o desempenho para observações que pertencem a regiões heterogêneas.

Os experimentos aqui apresentados, bem como os algoritmos, foram implementados em MATLAB, utilizando suas bibliotecas e funções padrões. Para os parâmetros do AHCP, foi considerado o valor de $p = 1$ para a tolerância de homogeneidade, ou seja, toda região com 99% ou mais de observações de mesma classe é considerada homogênea. Além disso, o limiar de separabilidade inicial escolhido foi de 0.5, que foi atualizado durante os experimentos conforme já descrito no capítulo 3.

Tabela 4.1: Confiança fora da amostra para os bancos sintéticos baseada nas regiões detectadas pelo ACPH (Algoritmo de Classificação por Particionamento Hierárquico). kNN foi executado usando $k = \sqrt{NúmerodeObservaçõesdeEntrada}$

Confiança Fora da Amostra dos Bancos Sintéticos

	Global (%)			Regiões Homogêneas (%)			Regiões Heterogêneas (%)		
Datasets	ACPH	SVM	KNN	ACPH	SVM	KNN	ACPH	SVM	KNN
Caso 1	88.8	95.0	95.6	97.5	96.6	97.5	64.3	90.5	90.5
Caso 2	85.6	85.6	85.6	100.0	100.0	100.0	78.3	78.3	78.3
Caso 3	77.5	82.5	81.3	100.0	100.0	100.0	75.3	80.8	79.5
Caso 4	80.6	82.5	80.6	93.6	89.4	85.1	75.2	79.7	78.8

Percebe-se pela tabela 4.1 que, na maioria dos casos, o ACPH teve valores de confiança similares aos obtidos com kNN e SVM. A maior discrepância pode ser notada no caso 1, para o qual o ACPH teve menor confiança que os demais algoritmos de classificação. No entanto, nesse caso, tanto o ACPH quanto o kNN tiveram a mesma acurácia em regiões homogêneas. Já no caso 4, o ACPH teve melhor desempenho que o kNN nos mesmos tipos de regiões.

Resultados similares foram obtidos quando avaliado o desempenho para os bancos de dados do UCI, como pode ser visto na tabela 4.2. Desses, apenas dois retornaram regiões heterogêneas: Uso de Métodos Contraceptivos e Avaliação de Carros. Em ambos, o ACPH teve melhor acurácia em regiões homogêneas quando comparado aos algoritmos kNN e SVM, embora o último tenha obtido melhores resultados para o banco de Avaliação de Carros.

Tabela 4.2: Confiança Fora da Amostra para Bancos do UCI Considerando Regiões Encontradas pelo ACPH, Algoritmo de Classificação por Particionamento Hierárquico. kNN foi executado para $k = \sqrt{NúmerodeObservaçõesdeEntrada}$.

Confiança Fora da Amostra para Bancos do UCI usando Validação Cruzada

	Global (%)			Regiões Homogêneas (%)			Regiões Heterogêneas(%)		
Datasets	ACPH	SVM	KNN	ACPH	SVM	KNN	ACPH	SVM	KNN
Câncer de Mama - Diagnóstico	91.7	62.7	92.8	91.7	62.7	92.8	–	–	–
Câncer de Mama - Prognóstico	63.1	76.3	74.7	63.1	76.3	74.7	–	–	–
Hipertireoidismo	96.3	83.7	93.0	96.3	83.7	93.0	–	–	–
Uso de Métodos Contraceptivos	64.0	63.8	70.5	83.2	64.8	81.8	60.9	63.9	68.9
Avaliação dpe Carros	97.5	99.9	96.1	98.1	99.9	96.5	62.3	100	75.0
Creche	99.8	100	100	99.8	100	100	–	–	–

Os resultados apresentados nas tabelas 4.1 e 4.2 permitem concluir que o processo de segmentação realizado pelo ACPH é por si só essencial para analisar em que porção do espaço a classificação se torna mais difícil inclusive psra algoritmos já muito bem consagrados como o kNN e o SVM.

4.2 Interpretação Local da Classificação

Para melhor avaliaras vantagens do ACPH, é necessário analisar as regiões homogêneas e heterogêneas individualmente. A tabela 4.3 mostra todas as regiões heterogêneas encontradas ao executar o ACPH para os bancos de dados sintéticos. Na maioria dos casos, à medida em que a proporção da classe A se distancia de 50%, e, consequentemente, a proporção de uma das classes se torna consideravelmente alta, a acurácia obtida pelo ACPH se aproxima da acurácia obtida pelo kNN e pelo SVM. Por outro lado, à medida em que a proporção da classe A se aproxima de 50%, a acurácia do ACPH tende a ser pior quando comparado aos demais algoritmos.

Tabela 4.3: Análise de Regiões Heterogêneas: características e confiança fora da amostra considerando apenas pontos pertencentes a cada região. ACPH se refere ao algoritmo de classificação por particionamento hierárquico. kNN foi aplicado com $k = \sqrt{NúmerodeObservaçõesdeEntrada}$.

Análise de Confiança Fora da Amostra em Regiões Heterogêneas em Bancos Sintéticos

Bancos	Características da Região			Confiança Fora da Amostra (%)		
	Proporção Classe A Dentro da Amostra (%)	Número de Obs. Dentro da Amostra	Número de Obs. Fora da Amostra	ACPH	SVM	KNN
Caso 1	13.3	45	17	82.3	88.2	88.2
	32.4	34	2	0.0	100.0	50.0
	35.1	37	9	55.6	88.9	88.9
	51.5	33	10	40.0	90.0	100.0
	93.1	29	4	100.0	100.0	100.0
Caso 2	17.1	70	25	76.0	76.0	72.0
	70.1	107	29	75.9	75.9	72.4
	81.5	92	16	87.5	81.2	87.5
	87.1	101	28	82.1	78.6	82.1
	87.5	16	8	62.5	87.5	87.5
Caso 3	12.4	97	26	80.8	88.5	80.8
	14.6	48	13	46.1	46.1	46.1
	30.3	89	22	54.5	59.1	59.1
	47.2	144	35	68.6	85.7	82.9
	92.1	114	31	90.3	90.3	90.3
	96.8	63	13	100.0	92.3	100.0
	97.6	41	6	100.0	100.0	100.0
Caso 4	4.4	68	19	100.0	100.0	100.0
	22.4	58	16	62.5	75.0	62.5
	62.5	24	4	75.0	75.0	75.0
	65.4	52	15	66.7	66.7	66.7
	69.7	66	17	58.8	58.8	58.8
	70.0	50	14	71.4	92.9	100.0
	81.0	58	9	100.0	100.0	100.0
	87.8	49	15	66.7	66.7	66.7
	94.4	18	4	100.0	100.0	100.0

Esse resultado é consistente com s ideia de que as regiões com proporções de classe próximas a 50% são, de fato, regiões de maior confusão e usar apenas a proporção local para classificação pode não ser suficiente para classificar qualquer observação que pertença a essas regiões. Além disso, os resultados da tabela 4.3 também mostram especificamente as regiões em que os demais algoritmos têm mais dificuldade

em classificar, o que confere maior interpretabilidade não só ao método proposto, mas também a outros métodos de classificação que podem vir a ser utilizados em conjunto.

Como os bancos de dados sintéticos são bidimensionais, esses podem ser melhor analisados a partir de gráficos. Considerando a tabela 4.3 e as demais regiões homogêneas por ela não-representadas, as figuras 4.1,4.2,4.3 e 4.4 apresentam a proporção da classe A referente às diversas regiões encontradas pelo ACPH nos quatro casos sintéticos, em que regiões mais pretos têm alta proporção de classe A e regiões mais pretas têm alta proporção de classe B. Assim, uma observação que pertence a regiões mais alaranjadas, segundo as figuras 4.1,4.2,4.3 e 4.4, possuem probabilidades similares de ser classificada como A ou B e, portanto, pertencem a regiões de confusão e difícil classificação. Nesse caso, essas regiões alaranjadas são as mesmas regiões descritas pela tabela 4.3, logo essa tabela é de extrema importância na interpretação dos resultados em bancos de alta dimensão.

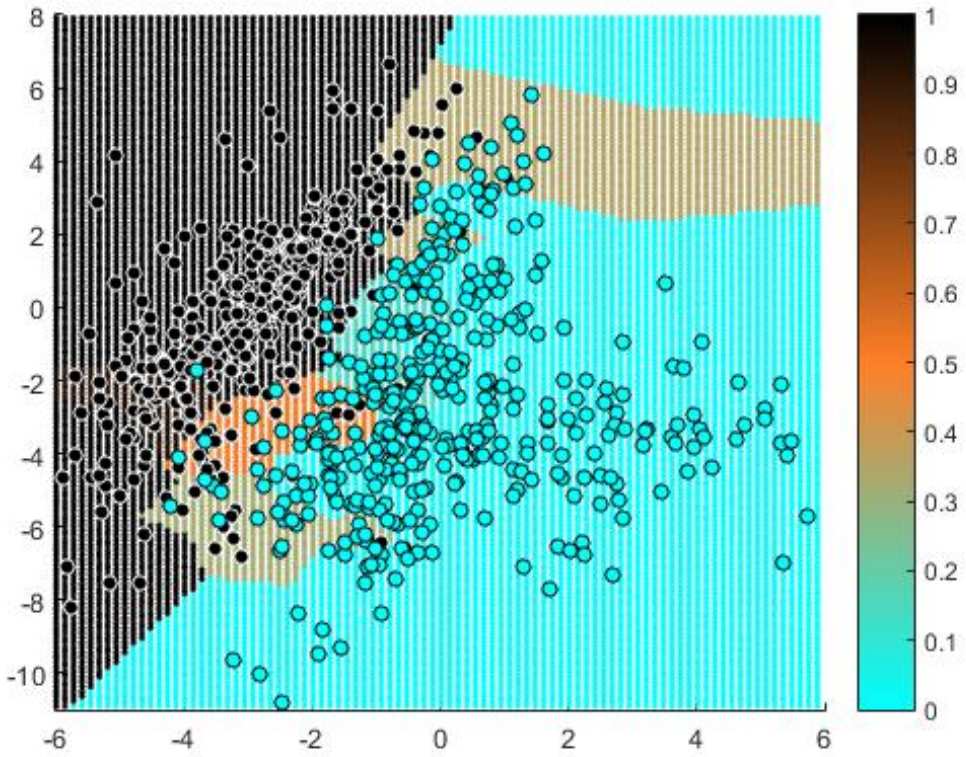


Figura 4.1: Representação visual das regiões encontradas para o caso sintético número 1. Aqui, a classe A é dada pelos pontos pretos e a proporção dessa classe é representada conforme a barra de cores.

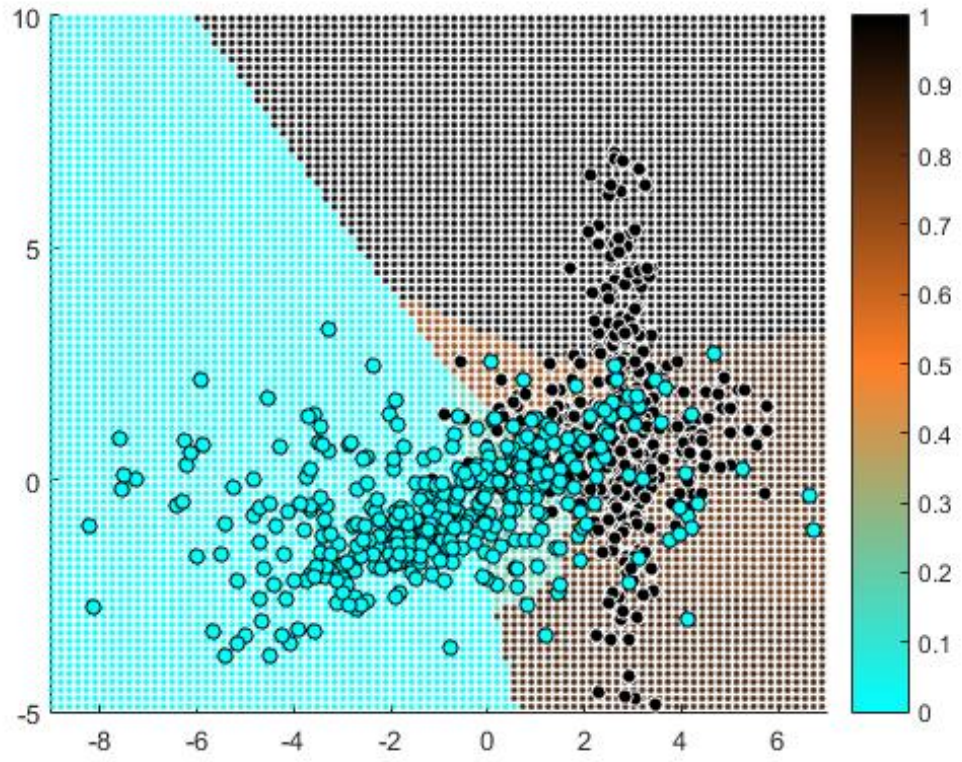


Figura 4.2: Representação visual das regiões encontradas para o caso sintético número 2. Aqui, a classe A é dada pelos pontos pretos e a proporção dessa classe é representada conforme a barra de cores.

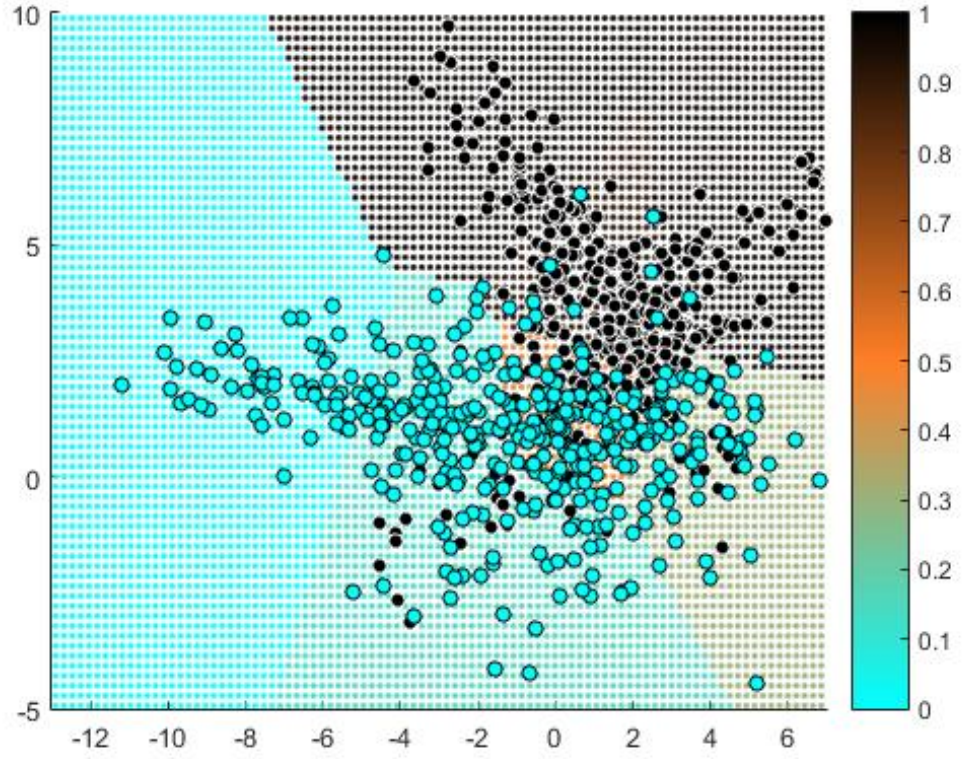


Figura 4.3: Representação visual das regiões encontradas para o caso sintético número 3. Aqui, a classe A é dada pelos pontos pretos e a proporção dessa classe é representada conforme a barra de cores.

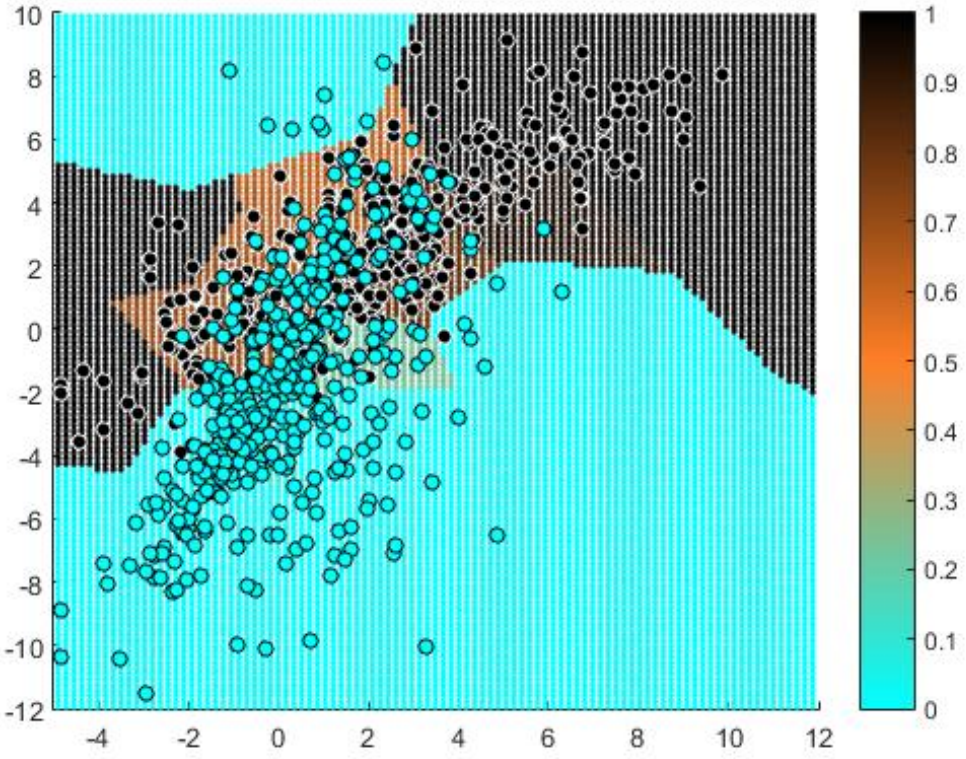


Figura 4.4: Representação visual das regiões encontradas para o caso sintético número 4. Aqui, a classe A é dada pelos pontos pretos e a proporção dessa classe é representada conforme a barra de cores.

Ademais, dadas as regiões encontradas pelo algoritmo aqui proposto, é razoável que se tente inferir a respeito dos valores dos atributos necessários para que uma observação pertença a uma região e não a outra. A tabela 4.4 apresenta as regiões e seus intervalos de valores de atributos para o caso 1 dos bancos sintéticos. O mínimo e o máximo de cada atributo em cada região permite maior conhecimento a respeito de uma nova observação sem que essa passe pelo processo de classificação. Nesse caso, considerando as regiões na tabela 4.4, se a observação tem atributo 1 maior que 1.57 e atributo 2 maior que -1.141, então ela certamente terá confiança de classificação maior que 51.5%.

Tabela 4.4: Descrição das regiões encontradas para o banco sintético Caso 1 considerando protótipo representante, classe majoritária local, confiança e mínimo e máximo de cada atributo.

Intervalos de Valores de Atributos das Regiões para o Caso 1

Protótipo da Região	Classe Local	Confiança (%)	Atributo 1		Atributo 2	
			Mínimo	Máximo	Mínimo	Máximo
(-2.599, -3.297)	A	51.5	-4.428	1.57	-4.592	-1.141
(0.112, 3.045)	B	64.9	-3.883	$+\infty$	1.804	$+\infty$
(-2.63, -5.783)	B	67.6	-4.428	$+\infty$	-4.592	$+\infty$
(-0.623, -0.818)	B	86.7	-3.883	$+\infty$	-4.592	$+\infty$
(-4.571, -2.323)	A	93.1	-4.428	$+\infty$	-4.517	$+\infty$
(-2.594, 0.968)	A	99.2	-2.042	$+\infty$	-0.957	$+\infty$
(3.056, -4.068)	B	100	0.435	$+\infty$	-5.892	$+\infty$
(-2.120, -9.349)	B	100	-3.666	$+\infty$	-9.591	$+\infty$
(-6.134, -6.973)	A	100	-14.322	$+\infty$	-7.382	$+\infty$
(-0.132, -3.858)	B	100	-1.313	$+\infty$	-26.331	$+\infty$
(-2.363, 4.877)	A	100	-31.433	$+\infty$	2.080	$+\infty$
(0.959, 0.315)	B	100	-0.237	$+\infty$	-6.472	$+\infty$
(-5.239, -3.999)	A	100	-4.987	$+\infty$	-4.770	$+\infty$
(1.469, 8.686)	B	100	-19.641	$+\infty$	5.496	$+\infty$
(-1.596, -4.695)	B	100	-2.669	1.308	-5.771	-3.467
(-3.045, -1.280)	A	100	-3.805	-1.533	-2.561	1.076
(-1.393, 2.694)	A	100	-2.939	$+\infty$	0.619	$+\infty$
(-4.238, -0.983)	A	100	-4.103	$+\infty$	-1.713	$+\infty$
(0.269, -1.736)	B	100	-0.661	$+\infty$	-5.872	$+\infty$
(-0.816, -6.899)	B	100	-1.842	$+\infty$	-6.551	$+\infty$
(0.579, 2.791)	B	100	-0.508	22.093	-5.301	3.490
(-0.554, -5.829)	B	100	-1.755	$+\infty$	-4.784	$+\infty$

É preciso atentar, entretanto, que essa informação permite verificar apenas a não-pertinência da observação em uma região e não é suficiente para determinar a que região especificamente ela pertence. Por exemplo, uma inferência ingênua consiste em assumir que ter o atributo 1 dentro do intervalo $[-4.428, 1.57]$ e o atributo 2 dentro do intervalo $[-4.592, -1.141]$ é suficiente para que uma observação pertença à primeira região listada em 4.4. No entanto, essa conclusão só seria verdadeira se

as regiões fossem retangulares, o que não ocorre. Na prática, o algoritmo de agrupamento k-médias divide o conjunto de observações em polígonos cuja quantidade de lados é dependente do número de grupos k e, nesse caso, as regiões, em geral, não são retangulares. Há alguns algoritmos capazes de retornar os vértices dessas regiões dados os respectivos protótipos, mas conhecê-los não permite a verificação imediata da pertinência da observação à região. Nesse sentido, conhecer os valores de mínimo e máximo para cada atributo se torna vantajoso na medida em que antecipa, em alguns casos, informações a respeito da confiança da classificação de novas observações.

Capítulo 5

Conclusões

Esta dissertação propôs um algoritmo de classificação que utiliza particionamento hierárquico supervisionado para classificação local de novas observações. Tem como principal objetivo discriminar regiões de fácil classificação e regiões de confusão. Trata-se de um algoritmo com passos intuitivos: uma região em que todos os pontos, ou quase todos, possuem a mesma classe é intuitivamente uma região de fácil classificação; enquanto regiões de difícil classificação a princípio, quando possuem distribuições de clases bem distintas, claramente possuem alguma subrregião homogênea para uma das classes.

A partir dos experimentos realizados, foi possível notar que o Algoritmo de Classificação por Particionamento Hierárquico (APCH) se mostrou bastante competitivo quando comparado a outros algoritmos comumente usados para esse tipo de aplicação, como o kNN e o SVM. Embora o kNN tenha apresentado melhor desempenho classificatório na maioria dos casos, o APCH não só apresentou acurácia e desempenho muito próximos, como também permitiu uma análise mais detalhada dos resultados fora da amostra. Enquanto o kNN retorna apenas informação quanto aos vizinhos das observações de entrada e o SVM retorna apenas o hiperplano de fronteira, o ACPH fornece, além da acurácia local, informações sobre todas as vizinhanças encontradas.

Para amostras com altas dimensões, a visualização e compreensão de suas distribuições são, em geral, complexas e podem exigir o uso de técnicas de redução de dimensionalidade. Nesses casos, em que gráficos são inviáveis, tanto o kNN quanto o SVM não conseguem prover informações além da classe estimada, ou proporção de vizinhos da observação no caso do kNN.

O algoritmo de classificação por particionamento hierárquico é capaz de identificar as regiões em que há mais confiança na classificação e regiões em que essa confiança é reduzida. Sua saída permite compreender e avaliar como os dados de entrada estão dispostos no espaço em relação às sobreposições das classes. Também permite extrair outros tipos de informação que enriquecem o processo de classi-

ficação, como por exemplo os intervalos de atributos e os valores de confiança de cada região. Além disso, seu processo intrínseco de agrupamento supervisionado permite analisar como outros algoritmos de clasificação se comportam localmente. Os resultados do kNN e do SVM em regiões homogêneas e heterogêneas só foram obtidos porque o ACPH forneceu a estrutura de regiões. Portanto, o algoritmo de classificação por particionamento hierárquico não só pode ser aplicado como classificador, mas também como uma ferramenta para avaliar outros métodos de maneira descritiva e detalhada.

Referências Bibliográficas

[1] KOIKKALAINEN, J., PÖLÖNEN, H., MATTILA, J., et al. "Improved Classification of Alzheimer's Disease Data via Removal of Nuisance Variability", *PLoS ONE*, v. 7, n. 2, pp. e31112, fev. 2012. ISSN: 1932-6203. doi: 10.1371/journal.pone.0031112. Disponível em: <http://dx.plos.org/10.1371/journal.pone.0031112>.

[2] SOMASUNDARAM, G., SIVALINGAM, R., MORELLAS, V., et al. "Classification and Counting of Composite Objects in Traffic Scenes Using Global and Local Image Analysis", *IEEE Transactions on Intelligent Transportation Systems*, v. 14, n. 1, pp. 69–81, mar. 2013. ISSN: 1524-9050, 1558-0016. doi: 10.1109/TITS.2012.2209877. Disponível em: <http://ieeexplore.ieee.org/document/6291788/>.

[3] BIN ZHANG, MARIN, A., HUTCHINSON, B., et al. "Learning Phrase Patterns for Text Classification", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 21, n. 6, pp. 1180–1189, jun. 2013. ISSN: 1558-7916, 1558-7924. doi: 10.1109/TASL.2013.2245651. Disponível em: <http://ieeexplore.ieee.org/document/6457440/>.

[4] WILLEMS, L. L., VANHOUCKE, M. "Classification of articles and journals on project control and earned value management", *International Journal of Project Management*, v. 33, n. 7, pp. 1610–1634, out. 2015. ISSN: 02637863. doi: 10.1016/j.ijproman.2015.06.003. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S026378631500099X>.

[5] BISHOP, C. M. *Pattern recognition and machine learning*. Information science and statistics. New York, Springer, 2006. ISBN: 978-0-387-31073-2.

[6] PERES, R., PEDREIRA, C. "A new local–global approach for classification", *Neural Networks*, v. 23, n. 7, pp. 887–891, set. 2010. ISSN: 08936080. doi: 10.1016/j.neunet.2010.04.010. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0893608010000936>.

[7] ABU-MOSTAFA, Y. S., MAGDON-ISMAIL, M., LIN, H.-T. *Learning from data: a short course*. S.l., AMLbook.com, 2012. ISBN: 978-1-60049-006-4. OCLC: 808441289.

[8] DUDA, R. O., HART, P. E., STORK, D. G. *Pattern classification*. 2nd ed ed. New York, Wiley, 2001. ISBN: 978-0-471-05669-0.

[9] GOU, J., ZHAN, Y., RAO, Y., et al. "Improved pseudo nearest neighbor classification", *Knowledge-Based Systems*, v. 70, pp. 361–375, nov. 2014. ISSN: 09507051. doi: 10.1016/j.knosys.2014.07.020. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0950705114002779>.

[10] PAN, Z., WANG, Y., KU, W. "A new general nearest neighbor classification based on the mutual neighborhood information", *Knowledge-Based Systems*, v. 121, pp. 142–152, abr. 2017. ISSN: 09507051. doi: 10.1016/j.knosys.2017.01.021. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0950705117300333>.

[11] CALVO-ZARAGOZA, J., VALERO-MAS, J. J., RICO-JUAN, J. R. "Improving kNN multi-label classification in Prototype Selection scenarios using class proposals", *Pattern Recognition*, v. 48, n. 5, pp. 1608–1622, maio 2015. ISSN: 00313203. doi: 10.1016/j.patcog.2014.11.015. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0031320314004853>.

[12] MUSAVI, M., AHMED, W., CHAN, K., et al. "On the training of radial basis function classifiers", *Neural Networks*, v. 5, n. 4, pp. 595–603, jul. 1992. ISSN: 08936080. doi: 10.1016/S0893-6080(05)80038-3. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0893608005800383>.

[13] HAYKIN, S. S. *Neural networks: a comprehensive foundation*. 2nd ed ed. Upper Saddle River, N.J, Prentice Hall, 1999. ISBN: 978-0-13-273350-2.

[14] CHENPING HOU, FEIPING NIE, DONGYUN YI, et al. "Discriminative Embedded Clustering: A Framework for Grouping High-Dimensional Data", *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 6, pp. 1287–1299, jun. 2015. ISSN: 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2014.2337335. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6867384>.

[15] ZHANG, Z., JIA, L., ZHANG, M., et al. "Discriminative clustering on manifold for adaptive transductive classification", *Neural Networks*, v. 94,

pp. 260–273, out. 2017. ISSN: 08936080. doi: 10.1016/j.neunet.2017.07.013. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0893608017301764>.

[16] LEITE, E. V. C. *Particionamento Dinâmico para Algoritmo Local de Classificação*. Dissertação de M.Sc., Universidade Federal do Rio de Janeiro, UFRJ.

[17] PRINCIPE, J. C. *Information Theoretic Learning*. Information Science and Statistics. New York, NY, Springer New York, 2010. ISBN: 978-1-4419-1569-6 978-1-4419-1570-2. Disponível em: <http://link.springer.com/10.1007/978-1-4419-1570-2>. DOI: 10.1007/978-1-4419-1570-2.

[18] KAMPA, K., HASANBELLIU, E., PRINCIPE, J. C. "Closed-form cauchy-schwarz PDF divergence for mixture of Gaussians". pp. 2578–2585. IEEE, jul. 2011. ISBN: 978-1-4244-9635-8. doi: 10.1109/IJCNN.2011.6033555. Disponível em: <http://ieeexplore.ieee.org/document/6033555/>.

[19] XU, D. *Energy, Entropy and Information Potential for Neural Computation*. Ph.D. thesis, University of Florida, Gainesville, FL, USA, 1999.

[20] PERES, R. T., ARANHA, C., PEDREIRA, C. E. "Optimized bi-dimensional data projection for clustering visualization", *Information Sciences*, v. 232, pp. 104–115, maio 2013. ISSN: 00200255. doi: 10.1016/j.ins.2012.12.041. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0020025513000339>.

[21] SILVERMAN, B. W. *Density estimation for statistics and data analysis*. N. 26, Monographs on statistics and applied probability. Boca Raton, Chapman & Hall/CRC, 1998. ISBN: 978-0-412-24620-3.

[22] SCOTT, D. W. *Multivariate density estimation: theory, practice, and visualization*. Second edition ed. Hoboken, New Jersey, John Wiley & Sons, Inc, 2015. ISBN: 978-0-471-69755-8.

[23] JENSE, R. *An Information Theoretic Approach to Machine Learning*. Ph.D. thesis, Faculty of Science, Department of Physics, University of Tromso, Tromso, Norway, 2005.

[24] JAMES, G., WITTEN, D., HASTIE, T., et al. *An Introduction to Statistical Learning*, v. 103, *Springer Texts in Statistics*. New York, NY, Springer New York, 2013. ISBN: 978-1-4614-7137-0 978-1-4614-7138-7. Disponível em: <http://link.springer.com/10.1007/978-1-4614-7138-7>. DOI: 10.1007/978-1-4614-7138-7.

Apêndice A

Artigo Submetido a Periódico

A Hierarchical Partitioning Algorithm for Classification

L.M. da Costa[a], E.V.C. Leite[a], R.T. Peres[b], C.E. Pedreira[a,*1]

[a]COPPE – PESC – Systems Engineering and Computer Science Program, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

[b]Mathematics Department (DEMAT), Federal Center of Technological Education of Rio de Janeiro (CEFET/RJ)

Abstract: We propose a new classification approach that takes advantages of both supervised and unsupervised learning. The aforesaid algorithm, we name Hierarchical Partitioning Algorithm (HPA), generates interpretable outputs in the sense that not only the class labels but also indications of certainty for different regions in the attributes space are provided. The HPA builds up partitions in an iterative way without the need of previous determining the number of prototypes representing these partitions. The k-means algorithm was employed as an unsupervised stage before applying a local supervised technique. The key idea is to cluster observations aiming to revel easy (and hard) to classify regions in the attributes space. The HPA has shown to be highly competitive when compared to commonly used classification algorithms like the kNN and the SVM, with the advantage of successfully selecting almost homogeneous regions while isolating regions where classification is difficult.

Keywords: Classification, Local-Global, Prototype, kNN, Cauchy-Schwarz, Clustering, Hierarchical Partitioning

[1]* Corresponding author.Corresponding address: UFRJ, COPPE-PESC, Av. Horácio Macedo, Centro de Tecnologia (CT), Bloco H, 3º andar, Ilha do Fundão, CEP 21941-914, Rio de Janeiro, RJ, Brazil.E-mail addresses: lygia.marina@gmail.com (L.M. da Costa), evcleite@gmail.com (E.V.C. Leite), rt.peres25@gmail.com (R.T. Peres), pedreira56@gmail.com (C.E. Pedreira)

Introduction

Classification methods may be seen through a Local-Global perspective. A pure global approach assumes that data is engendered by a phenomenon governed by a global fundamental law and does not take advantage of possible local generative structures (Peres and Pedreira, 2010). Global models aim to represent the entire space taken as a whole. In contrast, local models are a composition of sectional classification schemes that use specific subsets of the sample. Accordingly, when global methods are applied, the entire attribute space is equally treated, ignoring possible local characteristics in certain regions of this space.

One of the potential important benefits of introducing local information is the possibility of determining sub regions where the classifier is more (or less) likely to find the correct label. A local approach may take advantage of multiple local predictive models applied to different regions in the attribute space. Therefore, it is assumed that different groups of observations may be governed by different local distributions.

A well-known local tool that can be used for classifications the k-nearest neighbor (kNN) (Duda et al., 2001). It consists in using a set of k labeled observations that are neighbors of the observation one wish to label. A number of successful methods follow this conception, e.g. Gou et al. (2014). Several variations and improvements may be used to refine kNN ideas. For instance, in Pan et al. (2017), a scheme to select nearest neighbors based on a two-sided mode, produced interesting results for small sized datasets. In Calvo-Zaragoza et al. (2015), prototypes selection is deployed aiming to allow a faster kNN classification. Another example of local supervised method is Radial Basis Function Neural Networks (RBFN) (Musavi et al., 1992; Haykin, 1999), in which the set radial functions are spread in the attribute space. The labels are determined taking in account a measure of pertinence of the observation to each of those functions.

Some methods may take advantage from both, local and global approaches. Support vector machine (SVM) (Bishop, 2006), for instance, carries a global aspect in the sense that it generates a unique separator hyperplane, but the way this hyperplane is defined, is actually a local approach, since only a subset of observations is used. In Zhang et al. (2017), an approach combining unsupervised manifold learning, clustering and adaptive classification is proposed. Clustering is commonly used for unsupervised classification. This is done by partitioning the dataset into different subsets, the clusters, using similarity measures. The k-means (Bishop, 2006) is a quite popular clustering algorithm, whose main goal is to iteratively partition observations into k clusters such that each observation belongs to the cluster with the nearest distance to one of the k prototypes.

A local-global approach that combines supervised and unsupervised learning was proposed in Peres and Pedreira (2010). It starts by using unsupervised learning to divide the training data in local regions and then apply a supervised scheme in each of those regions independently. This approach takes advantage from the divide and conquer strategy, in which a complex problem is reduced to a set of much easier problems that are locally solved.

The k-means, or other clustering approach, may be employed as an unsupervised stage a before the application of a supervised technique. The idea is to cluster observations regardless of their class labels and then use a supervised algorithm to perform local classification within each cluster. This segmentation process may produce easy-to-classify clusters, and in the limit, it is possible to find homogenous regions with observations that belong to just one of the classes. On the other hand, the same process would be able to identify difficult-to-classify regions, what of course is also of interest.

In this article, we propose a new classification method that takes advantages from both supervised and unsupervised learning. It provides not only the estimated label for each observation but also an indication of certainty for different regions in the attributes space. Accordingly, besides providing the classification labels, the attribute space is partitioned and a sureness indicator is generated for each of these partitions. Each partition is represented by a prototype. Some of those demarcate almost homogenous regions where one may be well-nigh sure of the result. The proposed algorithm constructs the partitions in an iterative unsupervised way, without the need of previous determining the number of prototypes. In this manner, local and global information are combined to update the iterative process. Due to its iterative and hierarchical aspect, we named our method as Hierarchical Partitioning Algorithm (HPA).

Methodology

Let $\mathbf{x}=\{x_1, \ldots, x_N\}$ be a set of N d-dimensional observations to which a classification label, A or B, is assigned. Each of these observations correspond to d measured quantities we name attributes. The attribute space is quantized by a set of k prototypes $\mathcal{L} = \{\mathcal{V}_1, \ldots, \mathcal{V}_i, \ldots, \mathcal{V}_k\}$, where prototype $\mathcal{V}_i$ represents region $\mathcal{R}_i$, a subspace of the $\mathbb{R}^d$. We consider that any observation $x_j \in \mathcal{R}_i$ has a probability P_i to have a specific class label A and $(1 - P_i)$ to have class label B. Therefore, the attribute space is fragmented into regions, each of them represented by a prototype and with a correspondent probability to be assigned a class label A and B.

This fragmentation is iteratively performed by increasing the number of space subdivisions until it reaches a stopping criterion. For this purpose, we chose the k-means with an increasing k value, although other methods could be used for this

task. Two key questions rise at this point: whether there is an almost-homogeneous region $\mathcal{R}_i$ and whether there is a separable region $\mathcal{R}_i$. These terms will be defined in the sequence.

Definition: A region $\mathcal{R}_i$ is said to be almost-homogeneous if $(100 - p)\%$ of the observations $x_j \in \mathcal{R}_i$ are labeled as belonging to the same class for a chosen small p (e.g. $p < 10$).

Thus, a region $\mathcal{R}_i$ is almost-homogeneous, if most of its observations $x_j \in \mathcal{R}_i$ assume the prevailing class, A or B. As a characteristic of the proposed algorithm, an almost-homogeneous region is not re-divided, it is "reserved" and removed from the fragmentation process. The p parameter depends on how tolerant the user wants to be concerning homogeneity.

On the other hand, if the region is not almost-homogeneous, it is necessary to evaluate if there is still some informational gain in allowing this region to be re-divided. For this purpose, let Q_A be the estimated probability distribution function (pdf) of $\{\mathbf{x} \in \mathcal{R}_i,\ label\,(x) = A\}$ and Q_B the estimated pdf of $\{\mathbf{x} \in \mathcal{R}_i,\ \ label\,(x) = B\}$. Divergents (Cover and Thomas, 2006) may measure how a pdf diverges from a second pdf, it is a pseudo distance between two pdfs. Here, we apply the Cauchy-Schwarz divergence $\mathcal{D}_{r,q}$ between r and q defined as follows (Xu, 1999):

$$\mathcal{D}_{r,q} = -\log \frac{\left(\int r\,(x)\, q\,(x)\, dx\right)^2}{\int r^2(x)dx \int q^2(x)dx} \tag{A.1}$$

Definition: A region $\mathcal{R}_i$ is said to be separable if $\mathcal{D}_{A,B}$ between Q_A and Q_B is equal or greater than a chosen threshold (e.g. $s = 0.5$).

If the divergent is high enough, then we say that this region is separable in the sense that there is an advantage in allowing it to be re-divided. A high value of divergent indicates that there are parts of the region $\mathcal{R}_i$ in which there are considerable majority of points assuming the same class. Thus, keeping fragmenting allows revealing these sub-regions and test them as possible new almost-homogeneous ones. Otherwise, small values for the divergent suggest that the two classes are strongly overlapped and so further shattered will almost probably conduct to overfitting. Therefore, the divergent value indicates how local the space description should be.

Distributions r and q can be empirically estimated by using Parzen Window (Duda et al., 2001) with a multidimensional Gaussian function to obtain the estimated probabilities densities Q_A and Q_B. Let N_A and N_B be the number of

observations that belong to class A and B respectively. Then, each distribution can be estimated as follows:

$$Q_A = \frac{1}{N_A}\sum_{i=1}^{N_A} G_{0,\boldsymbol{I\sigma}^2}(x - x_i) Q_B = \frac{1}{N_B}\sum_{j=1}^{N_B} G_{0,\boldsymbol{I\sigma}^2}(x - x_j)\ , \quad x_i, x_j \in \boldsymbol{x} \tag{A.2}$$

Here, $G_{0,2\mathbf{I}\boldsymbol{\sigma}^2}(\mathbf{x})$ represents a Gaussian function with mean zero and $\mathbf{I}\boldsymbol{\sigma}^2$ as covariance matrix. Based on the window bandwidth rule-of-thumb (Silverman, 1998), for $\boldsymbol{\sigma}^2$ we considered a vector of estimated deviations for each class, considering standard deviation of each attribute and the observations dimension d, defined as below:

$$\begin{aligned}
\boldsymbol{\sigma}^2 &= \left\{ \left(\sigma_A^i\right)^2 + \left(\sigma_B^i\right)^2, \quad \forall \sigma_A^i \in \boldsymbol{\sigma}_A \quad and \quad \forall \sigma_B^i \in \boldsymbol{\sigma}_B \right\} \\
\boldsymbol{\sigma}_A &= std(\{\boldsymbol{x} \in \mathcal{R}_i, \quad label\,(x) = A\}) \times \left(\frac{4}{(d+2) \times N_A}\right)^{(d+4)^{-1}} \\
\boldsymbol{\sigma}_B &= std\,(\{\boldsymbol{x} \in \mathcal{R}_i, \quad label\,(x) = B\}) \times \left(\frac{4}{(d+2) \times N_B}\right)^{(d+4)^{-1}}
\end{aligned} \tag{A.3}$$

The terms of the divergence expression can be calculated using the estimated distributions as follows (Jense, 2005; Peres et al., 2013):

$$\begin{aligned}
\int r\,(x)\, q\,(x)\, dx = \frac{1}{N_A N_B}\sum_{i=1}^{N_A}\sum_{j=1}^{N_B} \int G_{0,\boldsymbol{I}}\,(x - x_i)\, G_{0,\boldsymbol{I}}\,(x - x_j)\, dx = \\
= \frac{1}{N_A N_B}\sum_{i=1}^{N_A}\sum_{j=1}^{N_B} G_{0,\boldsymbol{I}}\left(x_i - x_j\right) \\
\int r^2\,(x)\, dx = \frac{1}{N_A^2}\sum_{i=1}^{N_A}\sum_{j=1}^{N_A} G_{0,\boldsymbol{I}}\left(x_i - x_j\right) \\
\int q^2\,(x)\, dx = \frac{1}{N_B^2}\sum_{i=1}^{N_B}\sum_{j=1}^{N_B} G_{0,\boldsymbol{I}}\left(x_i - x_j\right)
\end{aligned} \tag{A.4}$$

Thus, the divergence $\mathcal{D}_{A,B}$ may be summarized as below (Peres et al., 2013; Xu, 1999):

$$\mathcal{D}_{A,B} = \frac{1}{N_A^2}\sum_{i=1}^{N_A}\sum_{j=1}^{N_A} G_{0,\boldsymbol{I}}\left(x_i - x_j\right) + \\ + \frac{1}{N_B^2}\sum_{i=1}^{N_B}\sum_{j=1}^{N_B} G_{0,\boldsymbol{I}}\left(x_i - x_j\right) - \\ - \frac{2}{N_A N_B}\sum_{i=1}^{N_A}\sum_{j=1}^{N_B} G_{0,\boldsymbol{I}}\left(x_i - x_j\right) \tag{A.5}$$

Note that the divergence depends only on the observations and on their class labels.

The stopping point concerning the space fragmentation is achieved when all $\mathcal{R}_i$ regions are almost-homogeneous or non-separable. At this point, the proportion of observations assuming class labels A are used to estimate the probability P_i for the region $\mathcal{R}_i$. The almost-homogeneous regions will then have $P_i \geq (100 - p)\,\%$ and the others will have $50\% < P_i < (100 - p)\,\%$.

The Hierarchical Partitioning Algorithm is summarized in figure A.1. The space segmentation represented by (b) is performed using k-means, starting with $K = 1$ repeated every time K is updated according to homogeneity and separability tests, as respectively in (c) and (i). Empirically, we defined as 'good almost-homogeneous subsets', the ones contain 5% or less observations labeled as the minority class.

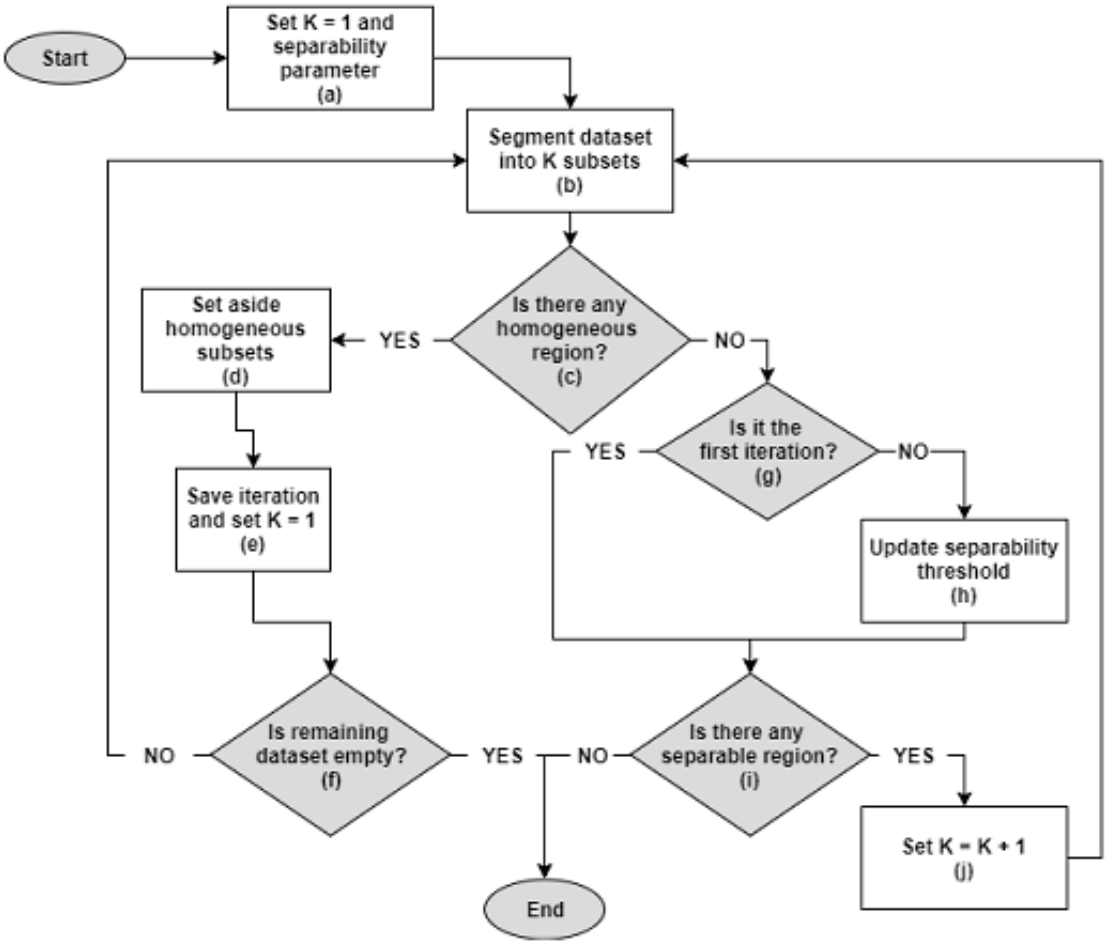


Figura A.1: Hierarchical Partitioning Algorithm (HPA) flowchart.

The processes (g) and (h) in A.1 are responsible for the separability threshold. We used $\mathcal{D}_{A,B} = 0.5$ as starting threshold. Throughout the training cycle this parameter is updated according to the error rate variation. Thus, for every cycle stage as A.1(b) is achieved, the current segmentation is used to classify the training data. The variation of the current misclassification rate relative to previous error rate is used to update the divergence threshold in A.1(h) as follows:

$$(\mathcal{D}_{A,B}\ threshold)_i = \frac{(\mathcal{D}_{A,B}\ threshold)_{i-1}}{\sqrt{(error\ rate)_i/(error\ rate)_{i-1}}} \tag{A.6}$$

In order to retrieve further information about the classifier, such as the segmentation hierarchy, all the iterations with almost-homogeneous subsets were saved, as well as the last iteration. The final one is reached when there is no more data to support space fragmentation or if there is not separable subset, as stated by (f) and (i) in A.1.

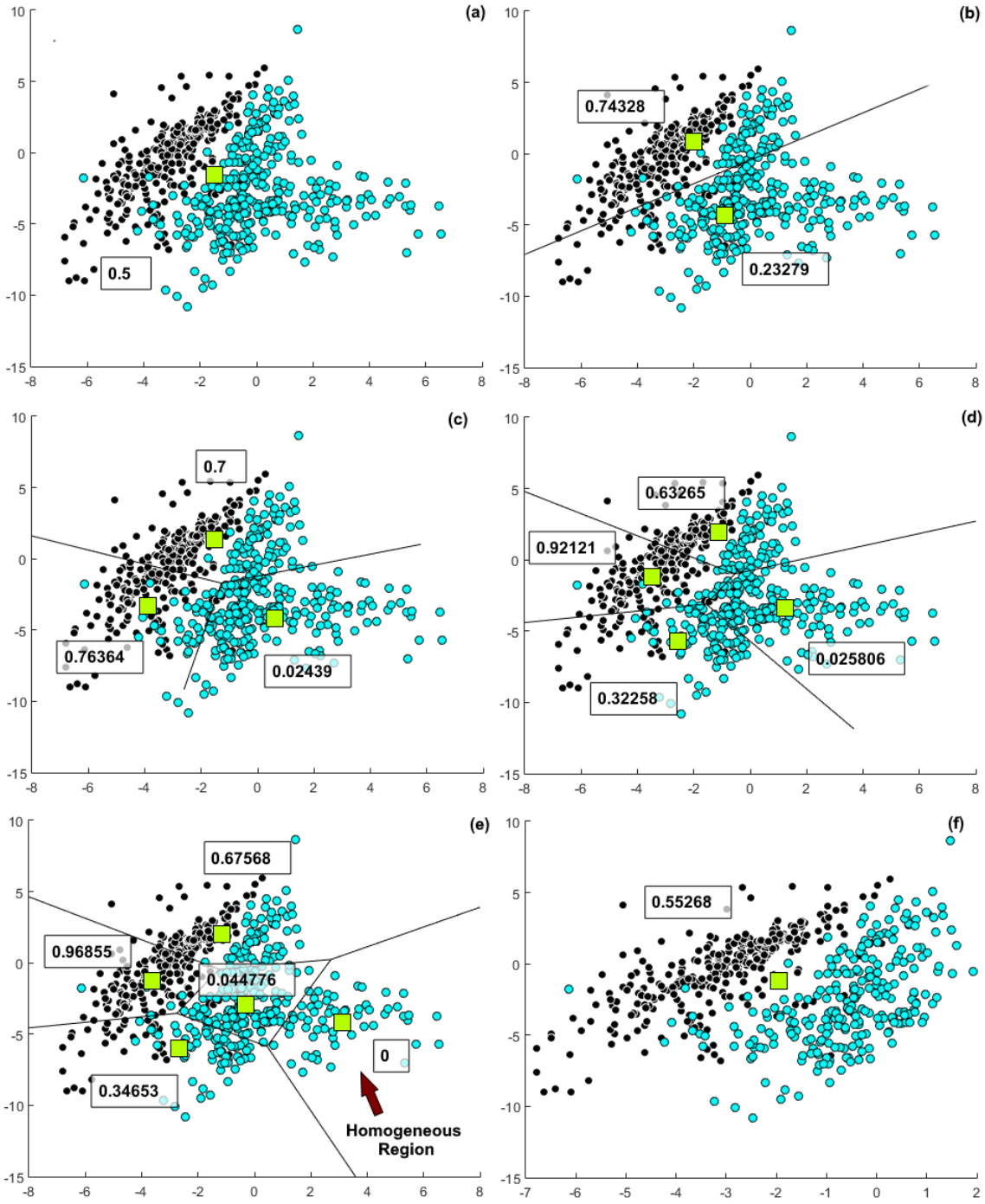


Figura A.2: HPA keeps segmenting the space until it reaches one or more almost-homogeneous regions. From (e) to (f) HPA removes the homogeneous region. The numbers in the rectangles are the proportion value for class 1, represented by blue markers. The green squares are the regions centroids.

The figure A.2 shows a sequence of iterations until the algorithm reaches an almost-homogeneous region with, in this case, 0% of class 1. Figure A.2(f) shows the training data after the data in the almost-homogeneous was put aside.

The sequence of iterations that ends with only non-separable regions is shown in figure A.3. The last plot in this figure represents the last iteration, in which all the regions have Cauchy-Schwarz divergence lower than the current threshold. These final and heterogeneous regions describe the space portion where there is an amount of uncertainty in classifying observations. In this case, any classification to be made is made using the local class proportion, based on the major class.

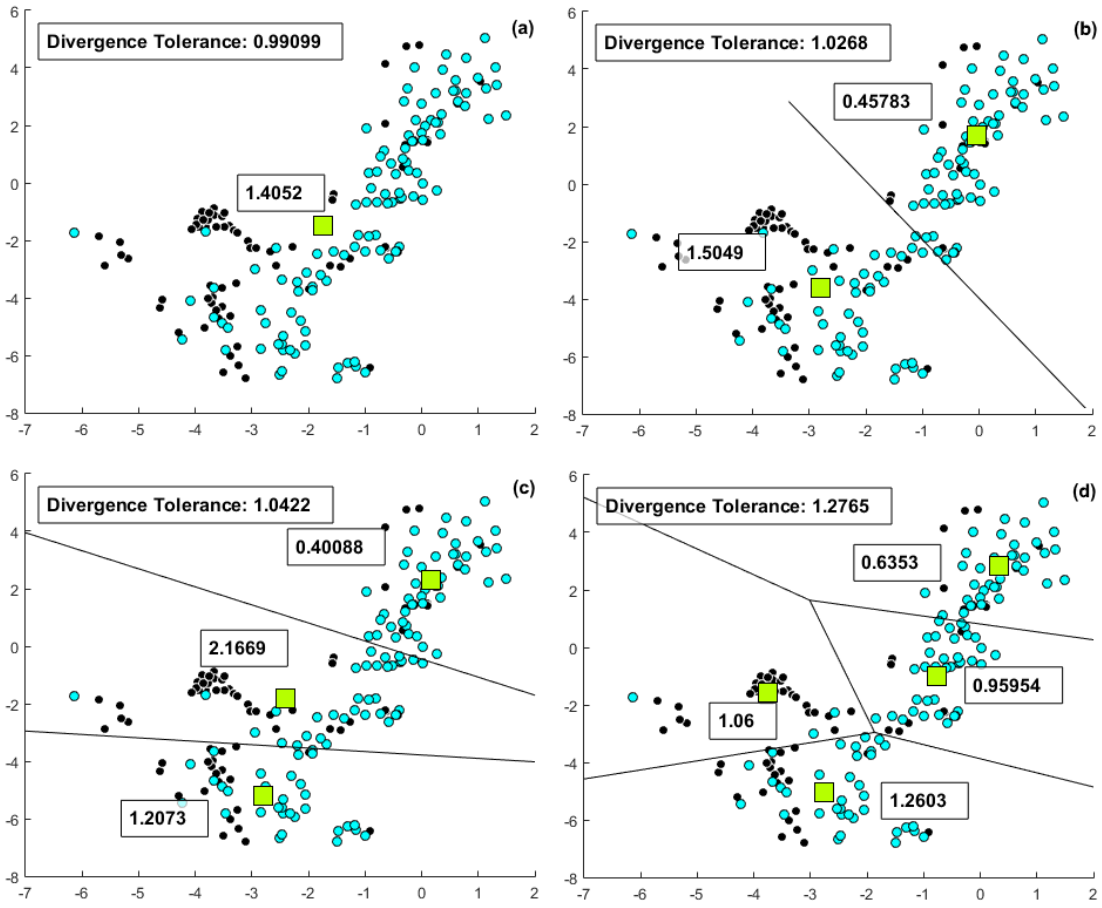


Figura A.3: Stopping criteria based on Cauchy-Schwarz divergence tolerance. The numbers inside the rectangles represent the Cauchy-Schwarz divergence of each region. If all the divergences are below the tolerance, then the algorithm has reached the stopping criteria.

To make sure that the local proportion value is somehow consistent with the local space, we performed a bootstrap test (Breiman, 1996). Bootstrapping is commonly used to estimate some statistical property from the dataset and it prevents overfitting. However, as it can be noted in Table 1, in most regions, the bootstrap standard deviation for the proportion of class 1 is lower than 10% and its mean is close to the original value. Thus, counting the frequency of class 1 is sufficiently

faithful to the local data.

Tabela A.1: Bootstrap for consistency test of local class proportion. No need of complex method to approximate the proportion value.

Local Proportion Consistency Test

Class 1 Proportion (%)	Bootstrap	
	Proportion Mean (%)	Proportion Standard Deviation (%)
13.3	13.2	5.6
32.3	30.4	7.5
35.1	33.8	6.3
51.5	50.2	8.4
93.1	93.1	5.3

Experiments

To analyze the method performance, several experiments were performed. The datasets consist of controlled generated data and UCI[2] repository datasets. The controlled datasets were produced using Gaussian mixture with random parameters, see A.4.

Concerning the UCI datasets, three of them are originally binary and the other three were transformed into binary. The Breast Cancer datasets and the Thyroid have two classes: positive or negative for the disease. The datasets known as Contraceptive Method, Car Evaluation and Nursery School had, initially, more than two classes, but there is a class that is much larger than the rest in all of three cases. In these cases, we considered two classes: the major class and a second class representing the rest of the original classes. Further information is shown in A.2.

The datasets 'Car Evaluation', 'Contraceptive Method' and 'Nursery School' have discrete attributes regarding to physical and personal features. Both Breast Cancer Diagnosis and Breast Cancer Prognostic data were obtained by feature extraction from breast mass digitalized images. In the Prognostic case, the dataset consists in follow-up data and the objective is to predict whether the cancer may recur or not. The Diagnosis dataset aims to classify if the mass is a malignant tumor or a benign one. Moreover, Thyroid Disease data has hormones values and the goal is to predict if there is hyperthyroidism or not. All these three datasets have continuous features.

[2] https://archive.ics.uci.edu/ml/index.php

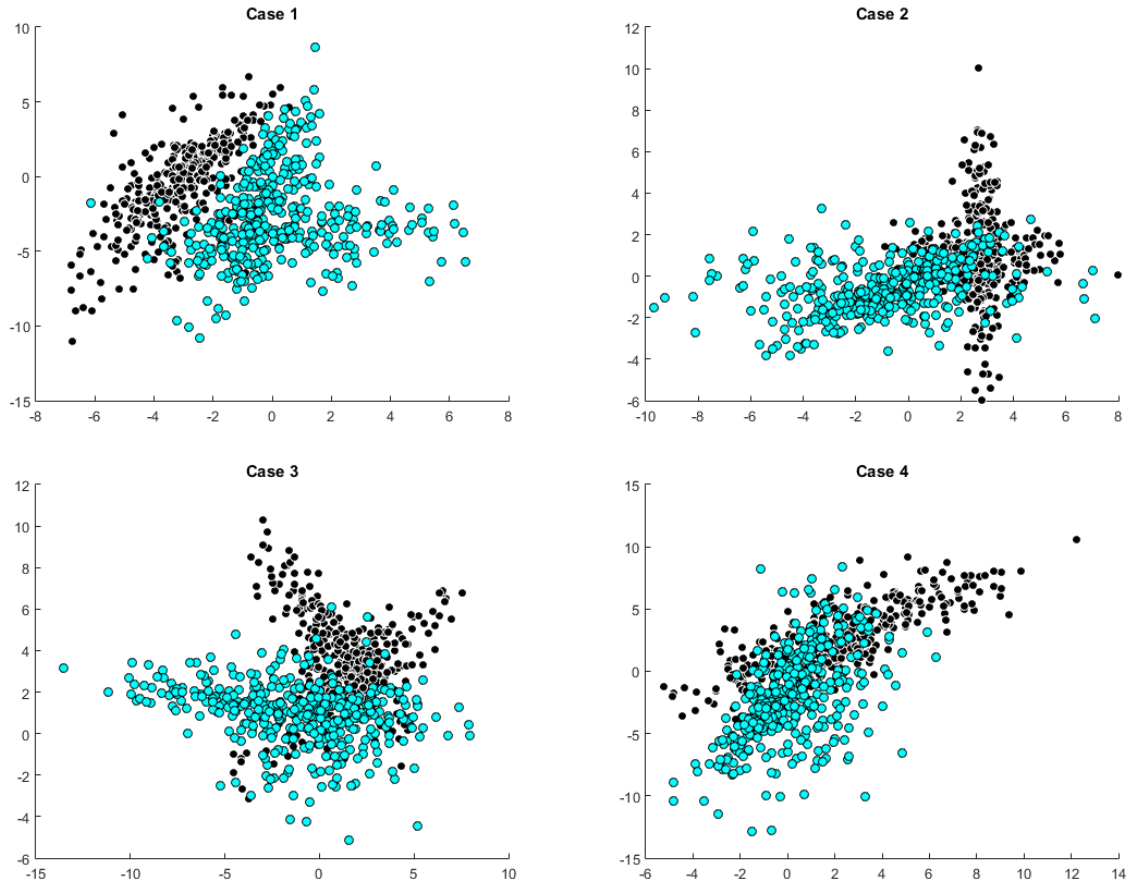


Figura A.4: Visual representation of synthetic cases that will be used for the experiments in tables A.3 and A.5

.

Tabela A.2: UCI Datasets information with class description and distribution.

UCI Datasets Information

Datasets	Attributes	Instances	Positive Classification (Class 1)		Negative Classification (Class 2)	
			Description	N. Obs.	Description	N. Obs.
Breast Cancer - Diagnosis	32	569	Malignant Tumor	212	Benign Tumor	357
Breast Cancer - Prognostic	32	198	Cancer Recurrence	47	Non-recurrence	151
Thyroid Disease	5	215	Hyperthyroidism	35	Not hyperthyroidism	180
Contraceptive Method	9	1473	Uses contraceptive methods	844	Does not use contraceptive methods	629
Car Evaluation	6	1728	Is at least acceptable	518	Is unacceptable	1210
Nursery School	8	12960	Is at least recommended	8640	Is not recommended	4320

Next section results are focused on out-of-sample accuracy rates, which were calculated based on an independent test sample for synthetic cases and on cross-validation for those from UCI. Due to the amount of available observations, we used leave-one-out cross-validation for Breast Cancer and Thyroid datasets. For Contraceptive Method, Car Evaluation and Nursery School datasets, we chose to perform k-fold cross-validation.

We compared the performance of the Hierarchical Partitioning Algorithm (HPA) against two widely used classification algorithms: Support Vector Machine (SVM) and k-Nearest Neighbors (kNN). All comparisons are based on the HPA final regions. Since the proposed algorithm has the advantage of providing not only an estimated class-label, but also an associated local uncertainty, the analysis was made considering three different situations: one comparing an overall accuracy rate, another considering accuracy rate for observations that belong to homogeneous regions and a third one considering accuracy in the heterogeneous regions. This way allows evaluation of both, global and local performances.

Both the kNN and SVM algorithms were executed using default parameters, with exception of the k value that specifies the number of neighbors to be considered for classification in kNN. Here, a rule of thumb popularized by Duda et al. (2001) was used and it defines k as being the square root of the amount of training observations: $k = \sqrt{Number\ of\ Insample\ Data}$.

Results

Results for synthetic cases and UCI datasets are shown in Tables A.3 to A.5. Note that, for overall results, high local accuracy regions are usually masked by those with low accuracy. Thus, comparing almost homogeneous and heterogeneous regions separately allows to confirm that HPA isolates regions in which classification may be difficult.

As it can be noted in table A.3, in most of the cases, HPA had similar overall accuracy rates when compared to kNN and SVM. The major discrepancy was noted on case 1, for which HPA had less accuracy than the benchmarks. However, in this case, both HPA and kNN had the same accuracy in homogeneous regions. In case 4, HPA performed better than kNN in the same type of regions.

Tabela A.3: Out-of-sample accuracy on synthetic datasets based on regions detected by HPA, which stands for Hierarchical Partitioning Algorithm. kNN was executed using $k = \sqrt{Number\ of\ Insample\ Data}$

Out-of-sample Accuracy on Synthetic Datasets

	Overall (%)			Homogeneous Regions (%)			Heterogeneous Regions (%)		
Datasets	HPA	SVM	KNN	HPA	SVM	KNN	HPA	SVM	KNN
Case 1	88.8	95.0	95.6	97.5	96.6	97.5	64.3	90.5	90.5
Case 2	85.6	85.6	85.6	100.0	100.0	100.0	78.3	78.3	78.3
Case 3	77.5	82.5	81.3	100.0	100.0	100.0	75.3	80.8	79.5
Case 4	80.6	82.5	80.6	93.6	89.4	85.1	75.2	79.7	78.8

Similar results were produced for UCI datasets, as it can be seen in A.4. Only two datasets ended up with heterogeneous regions: Contraceptive Method and Car Evaluation datasets. In both cases, HPA had better accuracy rates on homogeneous regions comparing to SVM and kNN, although SVM had the best performance for Car Evaluation dataset.

Tabela A.4: Out-of-sample accuracy on UCI datasets based on regions detected by HPA, which stands for Hierarchical Partitioning Algorithm. kNN was executed using $k = \sqrt{Number\ of\ Insample\ Data}$. Leave One Out Cross-Validation was used on Breast Cancer datasets and Thyroid dataset, while k-Fold Cross-Validation was used on Contraceptive Method, Car Evaluation and Nusery School datasets.

Out-of-sample Accuracy on UCI Datasets using Cross-Validation

	Overall (%)			Homogeneous Regions (%)			Heterogeneous Regions (%)		
Datasets	HPA	SVM	KNN	HPA	SVM	KNN	HPA	SVM	KNN
Breast Cancer - Diagnosis	91.7	62.7	92.8	91.7	62.7	92.8	–	–	–
Breast Cancer - Prognostic	63.1	76.3	74.7	63.1	76.3	74.7	–	–	–
Thyroid Disease	96.3	83.7	93.0	96.3	83.7	93.0	–	–	–
Contraceptive Method	64.0	63.8	70.5	83.2	64.8	81.8	60.9	63.9	68.9
Car Evaluation	97.5	99.9	96.1	98.1	99.9	96.5	62.3	100	75.0
Nursery School	99.8	100	100	99.8	100	100	–	–	–

Results in both Table A.3 and A.4 allow concluding that the segmentation process performed by HPA is by itself useful to analyze in which portion of the space other classification algorithms fail.

To better evaluate the HPA advantages, the heterogeneous regions should be individually analyzed. Table A.5 shows all heterogeneous regions reached when performing HPA for the synthetic datasets. In most of the cases, as class 1 ratio

distances from 50%, and consequently there is a high ratio for one of the classes, the HPA accuracy rate get closer to the rates of kNN and SVM. On the other hand, as class 1 ratio gets closer to 50%, the HPA accuracy rate tends to get worse comparing to the other algorithms.

This result is consistent with the idea that regions with similar to fifty-fifty class ratio are, in fact, confusion regions and a simple local ratio may not be enough to classify any observation that belong to these portions of the space.

Tabela A.5: Analysis of final heterogeneous regions of synthetic datasets: characteristics and out-of-sample accuracy considering only datapoints that belong to each region. HPA stands for Hierarchical Partitioning Algorithm. KNN was executed using $k = \sqrt{Number\ of\ Insample\ Data}$

Out-of-sample Analysis for Heterogeneous Regions on Synthetic Datasets

Datasets	Regions Characteristics			Outsample Accuracy Rate (%)		
	Insample Class 1 Ratio (%)	Number of Insample Data	Number of Outsample Data	HPA	SVM	KNN
Case 1	13.3	45	17	82.3	88.2	88.2
	32.3	34	2	0.0	100.0	50.0
	35.1	37	9	55.6	88.9	88.9
	51.5	33	10	40.0	90.0	100.0
	93.1	29	4	100.0	100.0	100.0
Case 2	17.1	70	25	76.0	76.0	72.0
	70.1	107	29	75.9	75.9	72.4
	81.5	92	16	87.5	81.2	87.5
	87.1	101	28	82.1	78.6	82.1
	87.5	16	8	62.5	87.5	87.5
Case 3	12.4	97	26	80.8	88.5	80.8
	14.6	48	13	46.1	46.1	46.1
	30.3	89	22	54.5	59.1	59.1
	47.2	144	35	68.6	85.7	82.9
	92.1	114	31	90.3	90.3	90.3
	96.8	63	13	100.0	92.3	100.0
	97.6	41	6	100.0	100.0	100.0
Case 4	4.4	68	19	100.0	100.0	100.0
	22.4	58	16	62.5	75.0	62.5
	62.5	24	4	75.0	75.0	75.0
	65.4	52	15	66.7	66.7	66.7
	69.7	66	17	58.8	58.8	58.8
	70.0	50	14	71.4	92.9	100.0
	81.0	58	9	100.0	100.0	100.0
	87.8	49	15	66.7	66.7	66.7
	94.4	18	4	100.0	100.0	100.0

Discussion and final remarks

The proposed algorithm, HPA, has shown to be highly competitive when compared to commonly used classification algorithms as the kNN and SVM. Its learning process was able of reserving almost homogeneous regions while isolating regions of

difficult classification is quite intuitive.

Although kNN presented a better performance in most of the cases, HPA not only had close accuracy rates, but also provided a more interpretable output. While kNN results carry only neighborhood information and SVM outputs a segregating hyperplane without local interpretation, HPA regions carry data distribution and local accuracy information. For samples with high dimension, visualizing and understanding its distribution may be complex and may require some dimension reducing algorithms. In cases like these, in which plotting is not feasible, both kNN and SVM fails to provide any further information besides class labels, or neighbors proportion in case kNN is being applied.

The HPA is capable of identifying the regions where there is a high certainty of classification and where there is not. Its output allows to understand and evaluate how the data is arranged in space concerning class mixtures. Also, the intrinsic supervised clustering of the proposed method allows to analyze how any other classification algorithm performs locally. The results of kNN and SVM in almost homogeneous and heterogeneous regions were only possible because HPA provided this structure. Thus, HPA may not only be used as a classification method, but also as a tool to evaluate other methods results in a descriptive and detailed way.

Acknowledgements

CEP: Rio de Janeiro State Research Foundation - FAPERJ, (Rio de Janeiro, Brazil), (grant: E26/201.189/2014)

CEP: Brazilian National Research Council - CNPq, (Brasilia, Brazil), (grant: 304243/2015-9)

LMC: National Council for the Improvement of Higher Education – CAPES, (Brasilia, Brazil)

References

Bishop, C.M., 2006. Pattern recognition and machine learning, Information science and statistics. Springer, New York.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140. doi:10.1007/BF00058655

Calvo-Zaragoza, J., Valero-Mas, J.J., Rico-Juan, J.R., 2015. Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. Pattern Recognit. 48, 1608–1622. doi:10.1016/j.patcog.2014.11.015

Cover, T.M., Thomas, J.A., 2006. Elements of information theory, 2nd ed. ed. Wiley-Interscience, Hoboken, N.J.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern classification, 2nd ed. ed. Wiley, New York.

Gou, J., Zhan, Y., Rao, Y., Shen, X., Wang, X., He, W., 2014. Improved pseudo nearest neighbor classification. Knowl.-Based Syst. 70, 361–375. doi:10.1016/j.knosys.2014.07.020

Haykin, S.S., 1999. Neural networks: a comprehensive foundation, 2nd ed. ed. Prentice Hall, Upper Saddle River, N.J.

Jense, R., 2005. An Information Theoretic Approach to Machine Learning (Ph.D. thesis). Faculty of Science, Department of Physics, University of Tromso, Tromso, Norway.

Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., Hummels, D.M., 1992. On the training of radial basis function classifiers. Neural Netw. 5, 595–603. doi:10.1016/S0893-6080(05)80038-3

Pan, Z., Wang, Y., Ku, W., 2017. A new general nearest neighbor classification based on the mutual neighborhood information. Knowl.-Based Syst. 121, 142–152. doi:10.1016/j.knosys.2017.01.021

Peres, R.T., Aranha, C., Pedreira, C.E., 2013. Optimized bi-dimensional data projection for clustering visualization. Inf. Sci. 232, 104–115. doi:10.1016/j.ins.2012.12.041

Peres, R.T., Pedreira, C.E., 2010. A new local–global approach for classification. Neural Netw. 23, 887–891. doi:10.1016/j.neunet.2010.04.010

Silverman, B.W., 1998. Density estimation for statistics and data analysis, Monographs on statistics and applied probability. Chapman & Hall/CRC, Boca Raton.

Xu, D., 1999. Energy, Entropy and Information Potential for Neural Computation (Ph.D. thesis). University of Florida, Gainesville, FL, USA.

Zhang, Z., Jia, L., Zhang, M., Li, B., Zhang, L., Li, F., 2017. Discriminative clustering on manifold for adaptive transductive classification. Neural Netw. 94, 260–273. doi:10.1016/j.neunet.2017.07.013